Bachelor's Degree course in Data Science and Engineering

Bachelor's Degree Thesis

Study of Manifold Geometry using Non-Negative Kernel Graphs.

Supervisors
Prof. Antonio Ortega
Prof. Javier Ruiz-Hidalgo

Candidate
Carlos Hurtado

Academic Year 2021-2022

Abstract

ENG: Given the increasing amounts of data being measured and recorded, effective dimensionality reduction systems have become necessary for a wide variety of tasks. A dataset can be characterized by its geometrical properties, including its point density, curvature, and dimensionality. In this context, the intrinsic dimension (ID) refers to the minimum number of parameters required to characterize a dataset. Many tools have been proposed for the estimation of ID, and the ones that achieve the best results are narrowly focused on solving this goal. These highly specialized estimators don't allow for the interpretation of the local geometry of the data in other aspects besides ID. Moreover, methods that do make this possible are not able to estimate ID reliably. We propose the use of non-negative kernel (NNK) graphs, an approach to graph construction that characterizes the local geometry of the data, to study the dimension and shape of data manifolds at multiple scales. We propose the use of a series of properties related to NNK graphs to gain insight into manifold datasets. In particular, we look at the number of neighbors in an NNK graph, the dimension of the low-rank approximations for both K-nearest neighbor (KNN) and NNK graphs, the diameter of the polytopes defined by NNK graphs, and the principal angles between the low-rank approximations of NNK graphs. Moreover, we study these properties at multiple scales using an algorithm that makes data sparse by merging points based on a choice of similarity. By using similarity based on local NNK neighborhoods we can subsample datasets preserving the geometrical properties of the initial dataset.

CAT: Amb l'augment de la mida de les dades, els sistemes efectius de reducció de la dimensionalitat s'han tornat necessaris per una gran varietat de tasques. Un conjunt de dades es pot caracteritzar per les seves propietats geomètriques, entre les quals es troben la densitat dels punts que hi té, la seva curvatura, i la dimensionalitat. En aquest context, la dimensió intrínseca (ID) fa referència al nombre mínim de paràmetres necessaris per caracteritzar un conjunt de dades. S'han proposat moltes eines per a l'estimació de DI, i les que aconsegueixen els millors resultats estan molt enfocades a resoldre aquest objectiu. Aquests estimadors altament especialitzats no permeten la interpretació de la geometria local de les dades en altres aspectes a part de la ID. A més, els mètodes que si ho permeten no són capaços d'estimar la ID de manera fiable. Proposem l'ús de grafs de kernel no negatiu (NNK), una aproximació a la construcció de grafs que caracteritza la geometria local de les dades, per estudiar la dimensió i la forma de les superfícies mutlidimensionals de dades a múltiples escales. Proposem l'ús d'una sèrie de propietats relacionades amb els grafs NNK per obtenir informació sobre diversos conjunts de dades. En particular, observem el nombre de veïns en un graf NNK, la dimensió de les aproximacions per anàlisi de components principals tant per als grafs K-nearest neighbor (KNN) com NNK, el diàmetre dels polítops definits pels grafs NNK i els angles principals entre les aproximacions per anàlisi de components principals dels grafs NNK. A més, estudiem aquestes propietats a múltiples escales utilitzant un algorisme que fa que les dades siguin més disperses fusionant punts en funció d'una tria de similitud. Utilitzant una similitud basada en els conjunts de veïns NNK, podem submostrejar conjunts de dades preservant les propietats geomètriques del conjunt de dades inicial.

ES: Con el aumento del tamaño de los datos, se han vuelto necesarios sistemas efectivos de reducción de dimensionalidad, útiles para una amplia variedad de tareas. Un conjunto de datos se puede caracterizar por sus propiedades geoemtricas, incluida su densidad de puntos, curvatura y dimensionalidad. En este contexto, la dimensión intrínseca (ID) se refiere al número mínimo de parámetros necesarios para caracterizar un conjunto de datos. Se han propuesto muchas herramientas para la estimación de la ID, y las que mejores resultados consiguen están estrechamente enfocadas a resolver este objetivo. Estos estimadores altamente especializados no permiten la interpretación de la geometría local de los datos en otros aspectos además de la ID. Asimismo, los métodos que hacen esto posible no pueden estimar la ID de forma fiable. Proponemos el uso de grafos de kernel no negativo (NNK), un enfoque para la construcción de grafos que caracteriza la geometría local de los datos, para estudiar la dimensión y la forma de las variedades de datos en múltiples escalas. Proponemos el uso de una serie de propiedades relacionadas con los grafos NNK para obtener información sobre múltiples conjuntos de datos. En particular, observamos el número de vecinos en un grafo NNK, la dimensión de las aproximaciones de rango bajo para los grafos K-nearest neighbor (KNN) y NNK, el diámetro de los politopos definidos por los grafos NNK y los ángulos principales entre las aproximaciones de bajo rango de los grafos NNK. Además, estudiamos estas propiedades en múltiples escalas usando un algoritmo que hace que los datos sean dispersos al fusionar puntos basados en una elección de similitud. Al usar una similitud basada en vecindarios NNK locales, podemos submuestrear conjuntos de datos conservando las propiedades geométricas del conjunto de datos inicial.

Contents

Chapter 1

Introduction

1.1 Motivation

Effective dimensionality reduction systems have become necessary for a wide variety of fields due to the increasing breadth of data collected thanks to the technological advances in the past years. The manifold hypothesis is defined as the hypothesis that high-dimensional datasets from real-world data tend to lie in low-dimensional manifolds contained in that high-dimensional space [23] [15]. This has led to the definition of *intrinsic dimension* (ID), which refers to the minimum number of parameters required to characterize a dataset while maintaining its structure. While multiple approaches have been proposed that achieve very good ID estimates on most datasets, these do not allow for any further interpretation of the local geometry of the data. Furthermore, those that do permit this to some degree, don't achieve good ID estimates.

Knowledge about ID is relevant for a wide range of contexts. For instance, many dimensionality reduction algorithms [60] [59] [56] require ID as an input parameter. Moreover, when using an autoencoder neural network architecture to learn a compressed representation of the data, the ID can be a good assessment of the size of the smallest hidden layer. Also, measuring the ID of features at different layers helps to understand how neural networks learn to transform data effectively [2], even when there is significant overparametrization. In addition, ID has also been shown to be a suitable descriptor to distinguish between different kinds of image structures [36]. Besides ID, other properties of a dataset can be of interest in certain tasks, such as the density of points in the manifold, or the manifold's curvature.

We, therefore, seek to develop an accurate estimator of ID for most datasets, that also makes it possible to gain insight into other geometrical properties of the data.

1.2 Project objectives

We tackle the problems described above by studying the local geometry of datasets using non-negative kernel (NNK) regression graph [54] properties. NNK is an approach to graph construction that has a geometric interpretation and is robust to the choice of the sparsity parameter. An NNK graph is built from an initial neighborhood such as a

KNN graph. Similarly to the orthogonalization step of orthogonal matching pursuits [55], NNK eliminates candidates that are geometrically redundant with candidates that have already been selected. The number of NNK neighbors characterizes the local geometry of the data.

We can derive a set of tools that can help in the understanding of manifold geometry using the properties of NNK neighborhood graphs. We first tackle the problem of ID estimation. Furthermore, while there is extensive literature on estimating the ID of manifolds, we have found little research on understanding their geometrical properties. We compare NNK neighborhoods at different scales to gain insight into the linearity or non-linearity of manifolds.

1.3 Organisation of this report

In Chapter 2, some of the basic concepts are introduced, such that all the information that is required to get the context and framework required to understand the work performed. In Chapter 3, the state-of-the-art and the most relevant work on the topic of research are presented. In Chapter 4 the set of tools we propose to tackle the problem we described is presented, together with two kinds of experiments: those that deepen our understanding of these tools, and those that use the methodology presented to explore the properties of manifold datasets. Finally, in Chapter 5, we summarize the main ideas presented and discuss future possible lines of work related to them.

Chapter 2

Background

2.1 Intrinsic Dimension

The ID provides critical information for understanding the structure of a dataset. ID was initially referred to as the minimum number of parameters that are required to represent the data while maintaining its structure, such that information loss is minimized, as defined by Bennett [5]. This concept has later been employed by Bishop in the context of neural networks [7], where it is stated that "a set in D dimensions is said to have an ID equal to d if the data lies entirely within a $d-$dimensional subspace of $\mathbb{R}^D$". Furthermore, some publications [46] [47] attempt to give a more precise definition to the concept of ID, and a recent literature review on the topic [14] concludes that "the prevailing ID definition views a point set as a sample set uniformly drawn from an unknown smooth (or locally smooth) manifold structure, eventually embedded in a higher-dimensional space through a nonlinear smooth mapping; in this case, the ID to be estimated is the manifold's topological dimension."

While there is consensus in the theoretical definition of ID, in practice some difficulties arise in its estimation when only a finite set of points is available. In Fig. 2.1 we show an example of how scale and point density affect the apparent ID. Another important consideration is the curse of dimensionality [4], which refers to the phenomena that arise when analyzing high-dimensional data. Mainly, as the dimension increases the volume of the space increases so rapidly that the available data becomes sparse, which causes most methods to underestimate the ID. Finally, some estimators become impractical as the number of points and their dimension increases due to their high computational complexity. This has led to the development of estimators of ID that take a wide range of approaches, most compromising on computational cost and interpretability. These approaches are discussed further in Section 3.1.

2.2 Graph Learning

The relations between the points in a dataset can be useful information in a wide variety of applications. A popular approach to convey this information is using a graph with edges based on a pairwise similarity metric. This metric can be as simple as the Euclidean

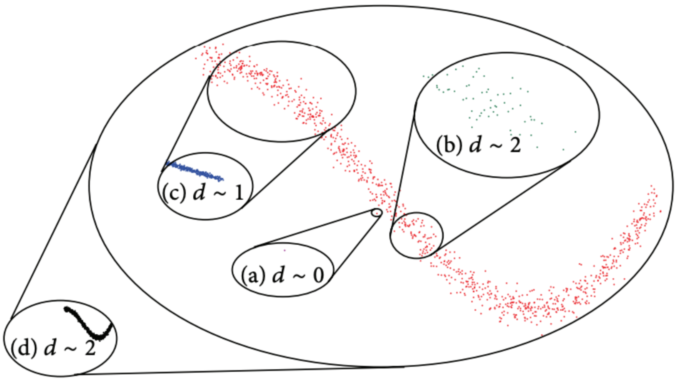

Figure 2.1. (a) At a very small scale the data looks zero-dimensional. (b) At a scale comparable to noise level, ID looks larger and in this case close to the embedding dimension. (c) At an intermediate resolution, the correct estimate of ID can be obtained. (d) At too large a scale the global dimension is obtained. Figure extracted from [14].

distance $\|x_i - x_j\|$, or much more complex, depending on the nature of the data.

Computing the chosen similarity metric on all pairs of points in a dataset would result in a fully connected weighted graph. There are many situations in which a much sparser graph is preferred, such that only the similarity between points that are "close enough" is preserved. We now discuss some approaches to achieving this.

2.2.1 Similarity-based approaches

Methods in this class sparsify the initial graph while seeking to maintain its properties. Weighted K-Nearest Neighbors (KNN) [21] and ε-neighborhood graphs (ε-graphs) [17] are among the most popular graph construction methods. These methods prune the initial graph while maintaining the weights of the edges that remain. Other approaches [30] [31] have been proposed in which the edge weights of the KNN/ε-graphs are optimized while preserving their connectivity (i.e., not removing any edges).

These approaches to the optimization of the similarity graph don't prune edges based on the geometry of the data, but rather on some threshold. A trade-off between bias and variance arises from this, and usually, the threshold parameters are chosen based on a heuristic approach or selected through cross-validation to optimize the performance on some task.

2.2.2 Locality inducing approaches

This family of methods starts from an initial set of neighbors, which can be derived from KNN/ε-graphs, and compute new edge weights that better reflect the data locality. An example of this is the local linear embedding (LLE) [51] algorithm, which solves:

$$\min_{\boldsymbol{\theta}:\boldsymbol{\theta}\geq 0} \|\boldsymbol{x}_i - \boldsymbol{X}_S\boldsymbol{\theta}\|_2^2\,, \tag{2.1}$$

where $\boldsymbol{X}_S$ corresponds to the matrix containing the features of the nearest neighbors S of $\boldsymbol{x}_i$. The solution $\boldsymbol{\theta}$ corresponds to the weights of the edges connecting $\boldsymbol{x}_i$ to its neighbors.

2.2.3 Non-Negative Kernel (NNK) regression graphs

A positive definite kernel $k(\boldsymbol{x_i}, \boldsymbol{x_j})$ corresponds to a transformation of points in $\mathbb{R}^d$ to points in a Hilbert space $\mathcal{H}$, such that similarities can be interpreted as dot products in this transformed space (generally referred to as *Kernel Trick*). This way, $k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi_i^T \phi_j$, where $\phi : \mathbb{R}^d \to \mathcal{H}$ and ϕ_i represents the transformed observation $\boldsymbol{x_i}$. The radial basis function (RBF) Gaussian kernel is an example of such a kernel, where σ refers to the bandwidth of the Gaussian curve over which the pairwise distances are weighted:

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(-\frac{||\boldsymbol{x_i} - \boldsymbol{x_j}||^2}{2\sigma^2}\right). \tag{2.2}$$

A KNN (or ε-graph) can be constructed by choosing the K largest inner products $\phi_i^T \phi_j$ (or those above a threshold ε). Then, a KNN (or ε-graph) corresponds to a sparse approximation of ϕ_i achieved by setting to zero the contributions of ϕ_j based on some threshold on $\phi_i^T \phi_j$.

Starting from an initial KNN/ε-graph neighborhood $\mathcal{S}$, NNK [54] selects an improved basis by solving for each node:

$$\boldsymbol{\theta}_{\mathcal{S}} = \min_{\boldsymbol{\theta}:\boldsymbol{\theta}\geq 0} \|\phi_i - \boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{\theta}\|_2^2\,, \tag{2.3}$$

where a linear combination, with weights given by $\boldsymbol{\theta}$, of the transformed neighbors $\boldsymbol{\Phi}_{\mathcal{S}}$ is used to approximate a vector ϕ_i in representation space. For similarities defined as inner products, the *Kernel Trick* can be used to rewrite (2.3) as:

$$\boldsymbol{\theta}_{\mathcal{S}} = \underset{\boldsymbol{\theta}:\boldsymbol{\theta}\geq 0}{\operatorname{argmin}} \frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{K}_{\mathcal{S},\mathcal{S}}\boldsymbol{\theta} - \boldsymbol{K}_{\mathcal{S},i}^T\boldsymbol{\theta}, \tag{2.4}$$

where $\boldsymbol{K}_{i,j} = k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$, and the i-th row of the adjacency matrix $\boldsymbol{W}$ is given by $\boldsymbol{W}_{i,\mathcal{S}} = \boldsymbol{\theta}_{\mathcal{S}}$ and $\boldsymbol{W}_{i,\mathcal{S}^c} = 0$.

NNK performs a selection similar to the orthogonal step in orthogonal matching pursuits [55] which makes NNK more robust to the choice of sparsity parameters in the initialization (i.e., K in KNN). Additionally, the resulting graph has a geometric interpretation [54] for the case of the Gaussian kernel 2.2, such that each edge in an NNK graph corresponds to a hyperplane with normal in the edge direction, as illustrated in Fig. 2.2. Points beyond each hyperplane are eliminated (edge weight zero).

NNK has been shown to deliver good results for semi-supervised learning, image representation [53], and label interpolation and generalization estimation in neural networks [52]. Furthermore, NNK has also been used to understand CNN channel redundancy [9] and to propose an early stopping criterion [10]. Moreover, graph properties have been also proposed for the understanding and interpretation of deep neural network performance [26], latent space geometry [37,38], improve model robustness [39]. The specific contribution of this work is the exploration of the effectiveness of the NNK algorithm

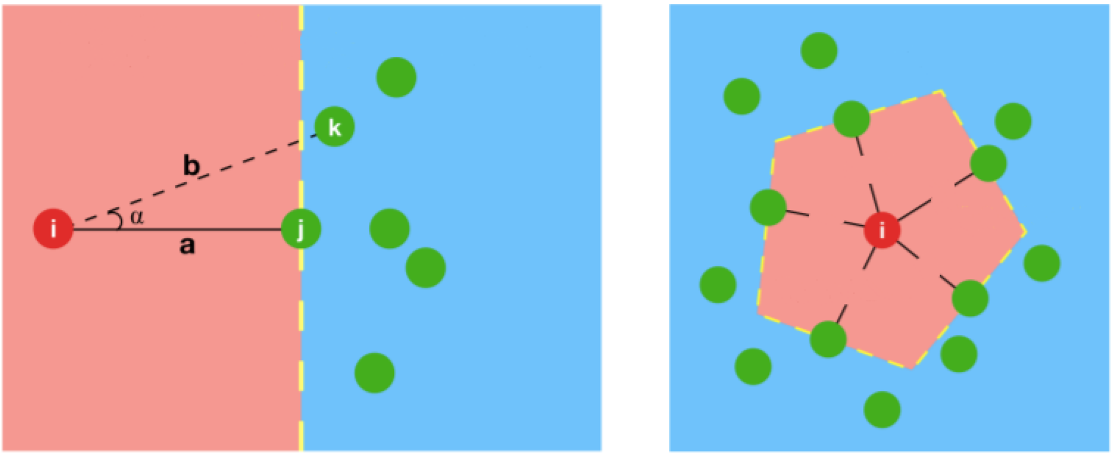

Figure 2.2. Local geometry (denoted in red) of the NNK graph construction for a Gaussian kernel. To the left, we show the hyperplane associated with an edge, beyond which points are ignored for the construction of the NNK graph. To the right, we show the convex polytope associated with a node that results from the hyperplanes associated with its NNK neighbors. Figure extracted from [54].

to get insight into the local geometry of the data, which can be useful in understanding the properties and structure of the whole dataset.

Chapter 3

Related Work

3.1 Intrinsic Dimension Estimation

ID estimators are designed under the assumption that every given dataset has been drawn from a smooth or locally smooth manifold which is contained in a high-dimensional space following a non-linear map. Additionally, each sample in the dataset is assumed to have been drawn independently and following a uniform distribution.

In this section, we introduce a taxonomy for ID estimators and discuss in more detail some examples that are relevant to our work. Following the taxonomy presented in a 2015 literature review and benchmark proposal [14], estimators can be classified as *projective*, *graph-based*, or *topological*. This taxonomy is different from others that have been proposed before [13], in that previous work classified methods as global or local. Global methods use all of the data and in local methods neighborhoods are analyzed separately. It is from the combination of local neighborhood information that an estimate for ID is obtained. This local approach is the most widely used in recent work, while the global approach has been abandoned.

In *Projective Estimators*, ID is quantified as the number of linearly independent vectors in the span of the projection subspace. These methods apply variations of Multi-Dimensional Scaling (the pairwise distances among data are preserved as much as possible), and principal component analysis (PCA) (finds projection subspace that minimized the projection error).

[24] [12] are two PCA-based approaches (i.e., projective estimators) in which intrinsic dimension is estimated first by partitioning the data into small neighborhoods and then applying PCA within each neighborhood, such that the ID is the number of eigenvalues that are larger than some threshold. These methods are shown [57] [33] [40] to depend heavily on the definition of the local neighborhoods as well as the threshold selection.

Later work [42] [32] builds on the technique of applying PCA locally by taking a multiscale approach in the neighborhood graph construction. The appropriate range of values for the graph constructions parameters, i.e., K for K-nearest neighbor (KNN) and ε for ε-neighborhood graphs (ε-graphs), such that the neighborhood is large enough that there are at least $K \geq ID$ neighbors, small enough that the manifold is linear and large enough so that the effects of noise are negligible.

While these multiscale approaches show promising results, the appropriate ranges for the neighborhood graph construction hyperparameters are highly sensitive to the density and distribution of points in the manifold [14]. Our proposed approach is robust to the choice of the neighborhood graph construction parameters and the resulting neighborhoods have a geometrical interpretation.

Graph-based Estimators construct a variety of graph structures based on the possible relationships between the sample points in the dataset. Insight into ID based on those graph structures is gained by leveraging knowledge in the field of graph theory.

In a notable publication [11] that follows this approach, the authors present an estimation method based on computing three statistics from graphs constructed from the data that give insight into its ID. From the K-Nearest Neighbor graph, the *reach* statistic S_1 is defined. A Geodesic Minimum-Spanning Tree (GMST, i.e., MST based on geodesic distances) is used to define the average degree S_2. Finally, from the K-Sphere of Influence Graph (kSIG, i.e., vertices are connected if their nearest-neighbor hyperspheres intersect.) the average number of neighbors shared by a given pair of points S_3 is obtained. They approximate each statistic by a Gaussian density function. Then, assuming equal probability for all possible intrinsic dimension values d_j, a posterior probability $P(d_j|S_n)$ is defined and used to arrive at an expected value of d.

Finally, in *Topological Approaches* for ID estimation, a locally smooth manifold embedded in a higher dimensional space is considered, such that we have a dataset with i.i.d. points sampled from the manifold through a smooth probability density function. Then, the topological dimension is the ID to be estimated. The most relevant type of topological estimators are *Nearest Neighbor-Based Estimators*, in which the ID of the data is described as a function of data neighborhood distributions.

In [48] a mathematical framework is presented such that the ID can be estimated from the distribution of the K nearest neighbors, upon which many other relevant topological nearest neighbor-based estimators are built. The authors acknowledge the limitations posed by the choice of a suitable value for K.

A very popular estimator in the literature is MLE [40], in which the authors assume the neighbors of each sample in the dataset to be events in a Poisson process. Then, [48] is used as an expression for the rate of the associated Poisson distribution, such that for each point, a maximum likelihood estimator is derived for d. These d values are then aggregated in some fashion to get a global estimate of the ID.

Two recent estimators that also rely on [48] are DANCO [16] and TwoNN [22]. In the former, statistics estimated on the data points with those estimated on uniformly drawn synthetic datasets of known ID are compared. The algorithm finds the d that minimizes the sum of the KL divergence applied to the distribution of the normalized nearest neighbor distances, and the distribution of pairwise angles. In the latter, ID is estimated from the distance between the first nearest neighbor B and the second nearest neighbor C for each point A in the sample. The d is estimated from the probability that B falls inside a hypersphere centered in A and with radius $r < d(A, C)$.

3.2 Intrinsic Dimension in Deep Learning

Deep Learning models have gotten progressively bigger, as have the datasets used to train these models. ID provides important insights into understanding the structure of these data, which ranges from work on adversarial robustness, neural network feature representations, assessing the difficulty of image datasets, objective function landscape analysis, and further understanding Self-Supervised Learning (SSL) models.

Some work has been done on improving our knowledge of adversarial attacks on Deep Neural Networks (DNNs) by using Local ID [3] as a measure of adversarial defence [44]. This approach to discriminating adversarial examples by analyzing the data structure through ID shows promising results. The same authors show an improvement in the generalization of models trained with noisy labels [45] by adapting the loss function according to the dimensionality of the deep representation subspaces during training. This is accomplished by leveraging the difference in learning in terms of representation dimensionality when using clean labels vs noisy labels.

In [45] as well as in [2] [50] it has been shown that the geometrical properties of the data representations in each layer of a DNN can be understood through their local intrinsic dimensionality. In the first layers of a neural network, ID increases, which relates to the early layers performing low-level pre-processing and feature extraction. These representations are task-independent and arise from features independent of the task. Later in the network, there is a dimensionality compression, a drop in ID, which is caused by the combination of multiple features in ways that make them relevant to the task. Furthermore, the ability of the model to compress the dimensionality of data representations in these last layers is indicative of the model's generalization [2]. Additionally, the analysis in [50] indicates that the representation manifolds learned by deep models usually are low-dimensional. This is said to be encouraged by the optimization process.

ID has also been used to estimate the dimension of popular natural image datasets, which are believed to lie in a low-dimensional manifold [23] [15], and assess its relation to deep learning models. In [49], the authors use a variant of MLE [40] to show that natural image datasets do indeed have a low ID relative to the high dimensional pixel representation. Moreover, DNNs are shown to generalize better on lower-dimensional datasets.

ID has also been used to better understand objective function landscapes [41]. In this context, the ID of an objective function is defined as “the lowest dimensional subspace in which one can optimize the original objective function to within a certain level of approximation error.” In other words, it is the smallest number of parameters required to find a satisfactory solution to the optimization problem. With this measure of ID, [41] quantitatively compares the difficulty of some supervised and reinforcement learning problems. As a byproduct of the minimization of the size of the objective function landscape, this approach to ID also allows for the compression of networks.

The above definition of ID [41] has been used to study the effectiveness of language model fine-tuning [1]. In this work, the authors show that ID on common NLP tasks is significantly lower on models pre-trained rather than on fully trained models. It is interpreted that during fine-tuning the model encodes the new task in terms of the pre-trained representations and in the process their minimal description length [28] is compressed.

Chapter 4

Methodology and Experiments

In this chapter, we describe our main contributions. In Section 4.1 we first derive an estimator of ID from NNK graphs and compare it to other state-of-the-art estimators. In Section 4.2 we describe other properties of NNK graphs that can be useful in the study of manifolds. Finally, in Section 4.3 we study the geometry of a wide range of manifolds by comparing the properties of NNK graphs at multiple scales. The code used for the experiments run in this section can be found in the following Github repository.

4.1 ID Estimation using NNK Graphs

4.1.1 Deriving ID from an NNK graph

For the Gaussian kernel, a hyperplane is associated with each edge in an NNK graph, with normal in the edge direction. Points beyond it are ignored for the construction of the graph. As a consequence, the local geometry of the NNK graph for a given node is a convex polytope around the node, where points outside the polytope are disconnected. For a sufficiently large number of initial neighbors, the local connectivity of an NNK graph will be a function of the local dimension of the manifold (see Fig. 4.1).

The number of neighbors in an NNK graph can be insightful, but it can vary locally based on (i) the distribution of the points sampled from a manifold and (ii) the location of the points relative to the geometry of the manifold (e.g., on the edges vs. the middle of the manifold). Besides, we can obtain information on the local geometry of the manifold by analyzing other properties of an NNK graph. Moreover, as we discuss in Section 4.2, we can also gain insight by comparing the properties of NNK graphs from different points in the same manifold.

As we discussed earlier in the literature review, an existing approach to estimate the ID of a manifold consists of performing a local parametrization by finding the local tangent plane in the neighborhood of a point and aggregating the estimated ID for each data neighborhood analyzed [24] [12] [42] [32]. In the case of noiseless samples from a linear subspace, PCA returns the local linear tangent space to the manifold. When dealing with noisy data, the scale at which we apply PCA, i.e., the points that fall into the window of observation, (see Fig. 2.1) must be small enough that we can ensure manifold linearity,

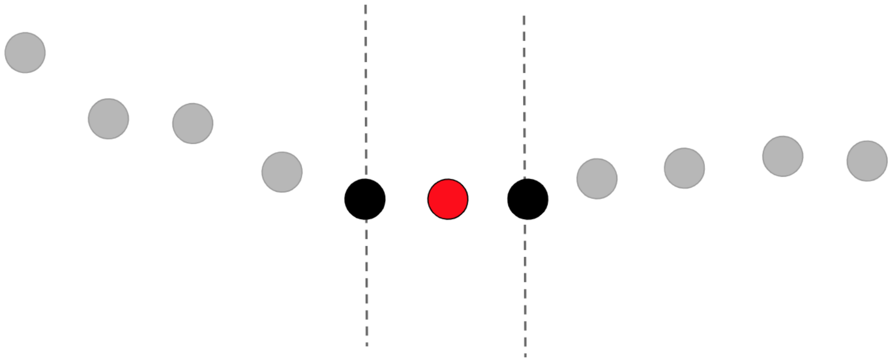

Figure 4.1. 1D manifold on a 2D space. A point's (red) neighbor (black) is selected in each meaningful direction, beyond which no other points (grey) are considered.

but large enough such that the manifold structure is discernible from noise.

Under the assumption that we are dealing with noiseless data, a low-rank approximation can be obtained from the NNK neighborhood vector subspace, such that the number of relevant principal components would be a robust estimate of the ID of the manifold, as depicted in Fig. 4.2. The concept of multi-scale analysis is addressed later on in the report.

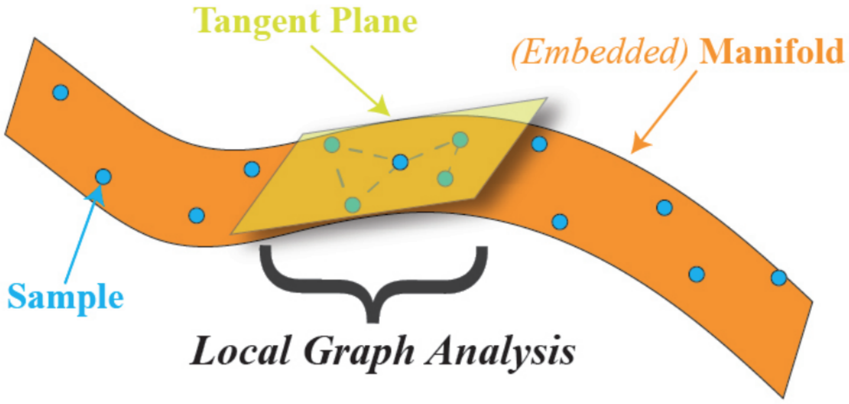

Figure 4.2. For a set of samples uniformly drawn from a locally smooth manifold, we can perform analyze a graph neighborhood, in this case, an NNK neighborhood, to obtain the plane tangent to the embedded manifold.

This way, assuming that a given dataset has been drawn from a smooth or locally smooth manifold which is contained in a high-dimensional space following a non-linear map and that each sample in the dataset has been drawn independently following a uniform distribution, we can estimate the global ID of a manifold by aggregating the local values of ID estimated from the NNK graph low-rank approximation dimensions, as described in Algorithm 1. The principal components chosen for the low-rank approximation are those with an eigenvalue above one-tenth of the largest eigenvalue [57]. From each low-rank approximation, we obtain a local estimate of ID. A global estimate of ID can be found by aggregating the local estimates. Assuming that there are more points in the middle of a manifold than on its edges, a global estimate of ID based on the median or the mode will result in a better approximation than the average of the local estimates.

Algorithm 1 NNK ID Estimation

```
Input:
    $X$: features
for each node $i = 1,2,...,N$ do
    $S_i = \{K \text{ nearest neighbors of node } i\}$
    $K_{S,S} = \text{RBF}(\sigma, X_S, X_S)$, $K_{i,S} = \text{RBF}(\sigma, X_i, X_S)$
    $W_{i,S} = \text{NNK}(K_{S,S}, K_{i,S})$, $W_{i,S^c} = 0$
    $S'_i = \{S_i : W_{i,S} > 0\}$
    $F_I = \text{PCA}(X_S - X_i)$
    $D_i = \{\text{count } \lambda_j : \lambda_j \geq 0{,}1 \cdot \lambda_{max}\}$
end for
$\hat{D} = \overline{D}$
Output: Intrinsic dimension estimate $\hat{D}$
```

The effectiveness of the NNK selection is governed by the sparsity parameter K used in the KNN search before the NNK step, and the bandwidth parameter σ since we are performing NNK on a Gaussian kernel similarity matrix. In the following sections, we discuss the impact of both parameters on the NNK graphs constructed from the data and their effect on the estimation of ID.

4.1.2 Adequate choice of K

In this section, we discuss how the size of the initial KNN neighborhood relates to the NNK graph, to find a meaningful range of values of K in terms of the local geometry of the data. By definition, NNK is robust to the choice of the sparsity parameter K used to build the initial graph. This is because the number of neighbors that are assigned non-zero weights is not predetermined and instead depends on the geometry of a point's neighborhood. Experimental results [54] suggest that the edge density of an NNK graph tends to saturate to a constant as we increase K. By requiring only neighbors that help in the representation of a node in similarity space, NNK graphs enforce sparsity.

It is also the case that K has to be large enough that a point can be reconstructed from its initial neighborhood. We previously argued that the number of points in an NNK neighborhood will be a function of the local dimension of the manifold. The rate at which the connectivity changes in relation to the ID of the data is unclear, and we look further into this in the following experiment.

Experiment

To better understand the relationship between ID and the size of an NNK neighborhood, we run the NNK graph construction algorithm on a handful of datasets with known ID. We perform this experiment on the synthetic datasets used in the ID estimator benchmark proposal [14]. In particular, we use a tool proposed by [27], which allows us to generate synthetic datasets by uniformly drawing samples from 13 manifolds of known ID linearly or nonlinearly embedded in higher dimensional spaces. The datasets suggested by the benchmark, generated using the aforementioned tool, are described in Table 4.1.

Dataset name	Description	**N**	**d**	**D**
$\mathbf{M}_1$	10-dimensional sphere linearly embedded	2500	10	11
$\mathbf{M}_2$	Affine space	2500	3	5
$\mathbf{M}_3$	Concentrated figure, mistakable with a 3-dimensional one	2500	4	6
$\mathbf{M}_4$	Nonlinear manifold	2500	4	8
$\mathbf{M}_5$	2-dimensional helix	2500	2	3
$\mathbf{M}_6$	Nonlinear manifold	2500	6	36
$\mathbf{M}_7$	Swiss-Roll	2500	2	3
$\mathbf{M}_9$	Affine space	2500	20	20
$\mathbf{M}_{10a}$	10-dimensional hypercube	2500	10	11
$\mathbf{M}_{10b}$	17-dimensional hypercube	2500	17	18
$\mathbf{M}_{10c}$	24-dimensional hypercube	2500	24	25
$\mathbf{M}_{10d}$	70-dimensional hypercube	2500	70	71
$\mathbf{M}_{11}$	Möebius band 10-times twisted	2500	2	3
$\mathbf{M}_{12}$	Isotropic multivariate Gaussian	2500	20	20
$\mathbf{M}_{13}$	1-dimensional helix curve	2500	1	13

Table 4.1. Synthetic datasets suggested by the benchmark [14]. **N** is the dataset number of samples, **d** is the ID, and **D** is the embedding space dimension.

The data consists of 10 datasets with an ID between 1 and 10, and 5 datasets with an ID above 10. For this experiment, we have chosen to use the datasets with an ID less than or equal to 10. The NNK graphs have been constructed with an initial KNN neighborhood of size $K = 100$. This would be large enough, except for the largest ID datasets, for NNK to find enough relevant neighbors if two were needed in each dimension. The bandwidth parameter σ in the Gaussian kernel step in NNK has been defined as one-third of the average 15^{th} neighbor distance. While no heuristic choice of σ can ensure good conditioning of the similarity matrix, as shown in Section 4.1.3, defining the bandwidth based on distances in the neighborhood tends to work well in the NNK optimization.

Fig. 4.3 shows the average number of NNK neighbors for each dataset as a function of their ID, together with the 2^n curve. We can observe that indeed the number of neighbors in an NNK neighborhood grows exponentially with the ID of the underlying manifold. Moreover, the number of NNK neighbors is independent of the embedding space dimension, and it is the ID of the underlying manifold that dictates the size of the neighborhood. Such is the case for the $\mathbf{M}_6$ and $\mathbf{M}_{13}$, which are 6 and 1-dimensional

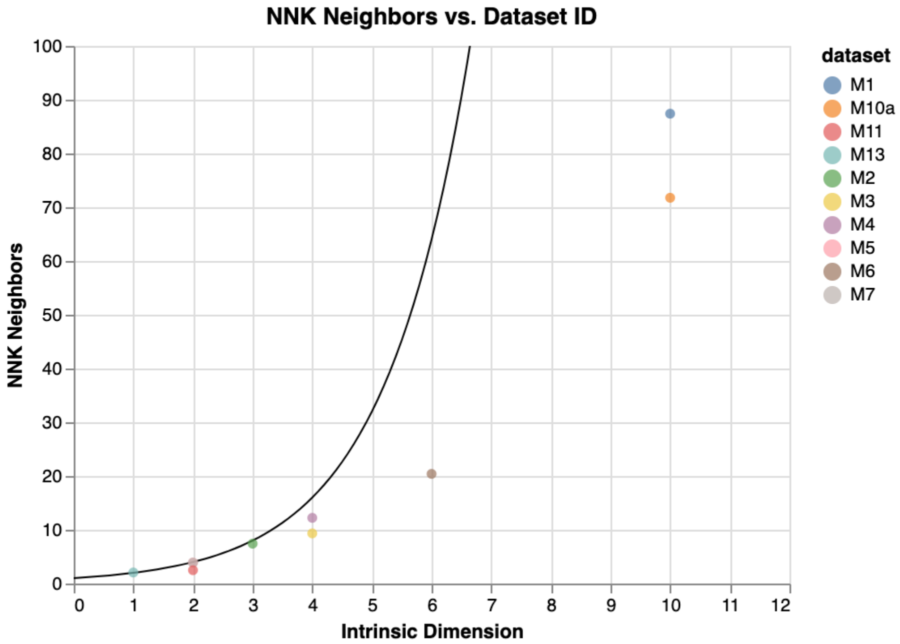

Figure 4.3. Scatterplot of the average number of NNK neighbors on 10 datasets consisting of points sampled from manifolds of known ID embedded in higher dimensional spaces. We also show a line with the 2^n curve. It can be observed that the number of NNK neighbors selected from an initial KNN graph of size $K = 100$ grows exponentially with a manifold's ID.

manifolds embedded in a 36 and 13-dimensional space respectively.

On the one hand, this finding gives a positive outlook on the geometrical properties of NNK graphs. On the other hand, it can be seen that, as with most ID estimators, NNK is affected by the curse of dimensionality, in that the number of points necessary to capture the meaningful directions in an NNK neighborhood of high-dimensional manifolds grows exponentially with the manifold's ID. We expect this finding to be reflected in the estimates performed on the benchmark dataset (Table 4.1) in Section 4.1.4.

4.1.3 Adequate choice of σ

The bandwidth parameter σ provides flexibility to adapt to the local data distribution. There is no straightforward way to define the bandwidth parameter such that its value fits the local data distribution. The most typical approach is to define σ based on some heuristic on the distances in the dataset.

We compute the kernel matrices $K_{S,S}$ and $K_{i,S}$ for the neighborhood of each point. For this process, we can use a global approach to defining σ_{global}, which would be the same for all the neighborhoods, or we can instead define a local σ_{local} based on the local data distribution, such that the bandwidth is unique to each neighborhood. Furthermore, a combination of the two can also be possible. The differences between each approach will become more relevant in Section 4.3.2. In the context of ID estimation, since we are working under the assumption that points in a dataset are drawn from a locally smooth manifold following a uniform distribution, there will not be a significant difference between these approaches given that data neighborhoods will be identically distributed.

Experiment

We can assess the quality of σ in relation to the data by observing (i) the coherence $\mu = \max_{j \in \mathcal{S}} k_{i,j}$ of $\mathbf{\Phi}_{\mathcal{S}}$, where the maximum coherence is that of the neighbor that is most similar to a given node and (ii) the conditioning of the matrix $\boldsymbol{K}_{S,S}$, for each node. The conditioning is defined by the reciprocal conditioning number $\mathcal{K} = \frac{\lambda_{min}}{\lambda_{max}}$, where λ_{min} and λ_{max} correspond to the smallest and largest eigenvalues of the matrix $\boldsymbol{K}_{S,S}$.

A choice of σ can be defined as appropriate when it results in a coherence that is neither too close to 0 nor 1, while at the same time maintaining the conditioning of the matrix used in the optimization. On a well-conditioned matrix, we would find $\mathcal{K} = 1$, and conditioning gets worse as $\mathcal{K}$ decreases.

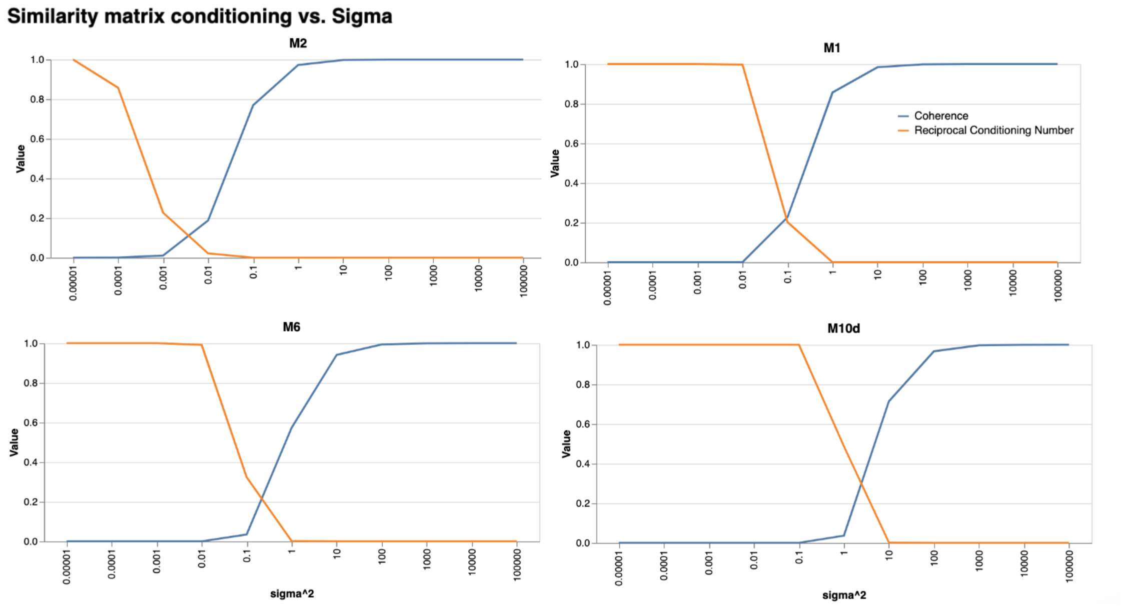

Figure 4.4. Average μ and $\mathcal{K}$ for different values of σ^2 on four datasets generated by uniformly drawing samples from manifolds (see Table 4.1 of known ID embedded in higher dimensional spaces. The datasets shown are $\mathbf{M}_2$ ($\mathbf{d} = 3$ and $\mathbf{D} = 5$), $\mathbf{M}_1$ ($\mathbf{d} = 10$ and $\mathbf{D} = 11$), $\mathbf{M}_6$ ($\mathbf{d} = 6$ and $\mathbf{D} = 36$), and $\mathbf{M}_{10d}$ ($\mathbf{d} = 70$ and $\mathbf{D} = 71$).

In this experiment, we explore the coherence μ and the reciprocal conditioning number $\mathcal{K}$ of the $\boldsymbol{K}_{S,S}$ matrices for different values of sigma on 4 of the synthetic manifolds described in Table 4.1, chosen so that we can observe the behavior of these metrics on a wide range of dimensions. We show these measures first using a fixed range of exponentially increasing values of sigma starting on $\sigma^2 = 10^{-5}$ and reaching $\sigma^2 = 10^4$. We later experiment on a range of bandwidth values defined from data, such that σ equals $\frac{1}{3}rd$ of the ith neighbor distance. This way, in the x-axis, the number of the neighbor the distance to which is used to define σ. Using this heuristic definition for σ we locate the ith neighbor 3 standard deviations away from the center node.

Figure 4.4 shows the trade-off between coherence and conditioning. Note that the range of values for σ at which we have a good balance between both metrics are different for each manifold since they are a function of the distances in the local neighborhoods. It can be seen that when σ is close to zero, all pairwise distances between nodes are also near zero, leading to a kernel similarity matrix $\boldsymbol{K}_{S,S}$ close to the identity matrix. In contrast, high values of σ correspond to a matrix with all similarities approaching 1, resulting in poor conditioning ($\mathcal{K} = 0$).

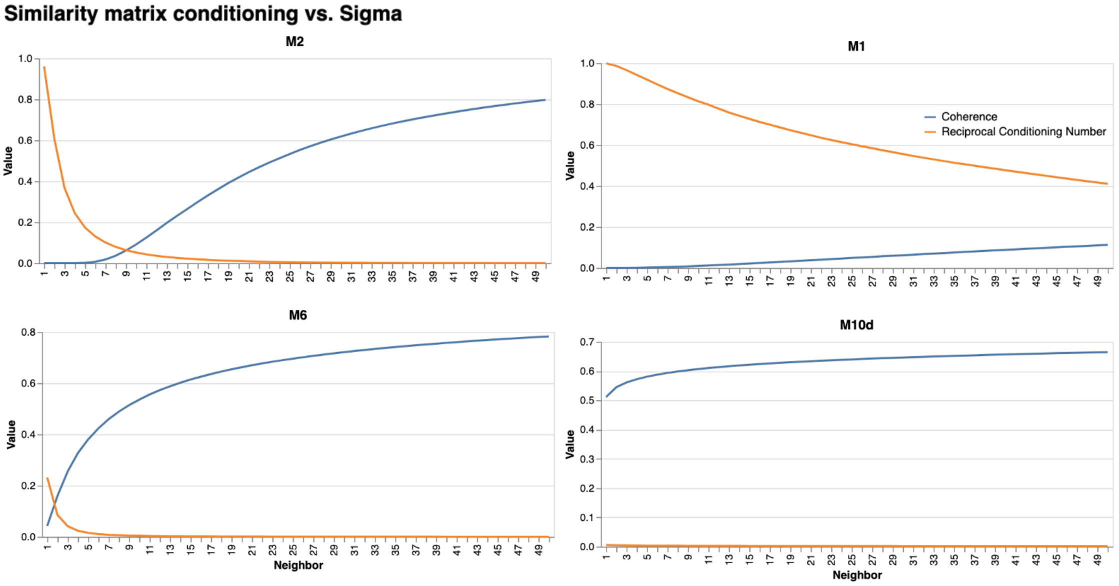

Figure 4.5. Average μ and $\mathcal{K}$ for different values of σ such that it equals $\frac{1}{3}rd$ of the ith neighbor distance, where the x-axis denotes the neighbor used in the calculation of σ.We show the results for four datasets generated by uniformly drawing samples from manifolds (see Table 4.1 of known ID embedded in higher dimensional spaces. The datasets shown are $\mathbf{M}_2$ ($\mathbf{d} = 3$ and $\mathbf{D} = 5$), $\mathbf{M}_1$ ($\mathbf{d} = 10$ and $\mathbf{D} = 11$), $\mathbf{M}_6$ ($\mathbf{d} = 6$ and $\mathbf{D} = 36$), and $\mathbf{M}_{10d}$ ($\mathbf{d} = 70$ and $\mathbf{D} = 71$).

Defining σ based on distances in the data tends to result (see Fig. 4.5) in good values for coherence and conditioning. Using $\frac{1}{3}rd$ of the ith neighbor distance, we seem to find a range of σ values with a balance between coherence and conditioning on some datasets, but this is not always the case. Moreover, as the dimension of the data increases, so do the distances between points and the heuristic does not seem to result in an appropriate value for the bandwidth.

4.1.4 ID estimation benchmark

We have compared the performance of our ID estimator to that of other state-of-the-art estimators, following a simplification of the guidelines of an ID benchmark proposal [14], such that we use a subset of the datasets proposed, and only one of the three evaluation metrics suggested is computed. We test the estimator based on NNK on the synthetic datasets described in Table 4.1, as well as on the real datasets described in Table 4.2, namely, ISOMAP and MNIST. The ISOMAP dataset consists of 698 gray-level images of size 64×64 of a face sculpture from different angles and lighting. The ID of this dataset is defined by its three degrees of freedom, these being two for the pose and one for the lighting direction. The MNIST dataset consists of 70000 gray-level images of size 28×28 showing handwritten digits. While the real ID of this dataset is not known, there is work proposing estimates for the different digits. The digit “1” is used in the benchmark, for which the proposed ID values are in the range $8 - 11$ [27] [19].

The results obtained from our estimator are compared to those obtained by other state-of-the-art estimators. We have included estimators for each of the categories described in the literature review. We have chosen BPCA [6] and MLSVD [43], which are projective estimators. In BPCA (Bayesian PCA), the appropriate dimension for the low-rank approximation is expressed as a maximum likelihood solution. MLSVD applies Singular Value Decomposition (SVD) locally and in a multi-scale fashion. kNNG [18] is a graph-based estimator with which ID is estimated from a heuristic on the properties of KNN graphs on local neighborhoods. We use CD [25] and Hein [27] as examples of topological fractal estimators. These estimators are based on the concept that the volume of a d-dimensional hypersphere of radius r scales as r^d, so that counting the number of points in a neighborhood of radius r can in some way approximate the rate of growth. This rate of growth d would be the ID of the manifold. Hein is a variant of CD that introduces a kernel function to avoid the scale dependency present in the latter. The topological nearest neighbor-based estimators are MLE [40] and DANCo [16]. Both estimate ID from a maximum likelihood estimator based on the distribution of the distances in a data neighborhood. The methods differ in the choice of the choices of distribution and neighborhood.

The performance of the estimators is assessed using the mean percentage error (MPE), which summarizes all the estimations in a single value computed as:

$$\text{MPE} = \frac{100}{\#\boldsymbol{M}} \sum_{\boldsymbol{M}} \frac{|\hat{d}_{\boldsymbol{M}} - d_{\boldsymbol{M}}|}{d_{\boldsymbol{M}}}, \tag{4.1}$$

where $\#\boldsymbol{M}$ is the number of tested datasets, $\hat{d}_{\boldsymbol{M}}$ is the estimated ID for the dataset $\boldsymbol{M}$, and $d_{\boldsymbol{M}}$ is the real ID of the dataset. On datasets whose ID belongs to a range, we have computed the MPE using the mean value of the range. The results shown for the estimators in the benchmark are the average of the ID obtained from 20 instances of each dataset. Due to resource and time constraints, we show the results for a single instance of each dataset for the NNK estimator.

We use Algorithm 1 to estimate the ID of the different datasets, using a $K = 100$ and $\sigma = \frac{1}{3} \cdot d_{15\text{th NN}}$. The results of the ID estimates obtained for the synthetic datasets with

Dataset name	Description	**N**	**d**	**D**
$\mathbf{M}_{\text{ISOMAP}}$	Gray-level images, size 64 × 64, face sculpture	698	3	4096
$\mathbf{M}_{\text{MNIST1}}$	Gray-level images, size 28 × 28, hand-written digits	7000	8-11	784

Table 4.2. Synthetic datasets suggested by the benchmark [14]. **N** is the dataset number of samples, **d** is the ID, and **D** is the embedding space dimension.

Dataset	*d*	MLE	kNNG	BPCA	Hein	CD	DANCo	MLSVD	NNK
$\mathbf{M}_1$	*10*	9.10	9.98	5.45	9.45	9.12	10.09	**10.00**	**10.00**
$\mathbf{M}_2$	*3*	2.88	3.03	**3.00**	**3.00**	2.88	**3.00**	**3.00**	**3.00**
$\mathbf{M}_3$	*4*	3.83	3.82	**4.00**	**4.00**	3.23	**4.00**	2.08	**4.00**
$\mathbf{M}_4$	*4*	3.95	4.76	4.25	**4.00**	3.88	**4.00**	8.00	**4.00**
$\mathbf{M}_5$	*2*	1.97	2.06	**2.00**	**2.00**	1.98	**2.00**	**2.00**	**2.00**
$\mathbf{M}_6$	*6*	6.39	11.24	12.00	**5.95**	5.91	7.00	12.00	8.00
$\mathbf{M}_7$	*2*	1.96	2.09	**2.00**	**2.00**	1.93	**2.00**	2.35	**2.00**
$\mathbf{M}_{10a}$	*10*	8.26	10.21	5.20	8.90	8.09	9.86	**10.00**	**10.00**
$\mathbf{M}_{11}$	*2*	2.21	2.03	1.55	**2.00**	2.19	**2.00**	1.00	1.00
$\mathbf{M}_{13}$	*1*	**1.00**	1.07	5.70	**1.00**	1.14	**1.00**	**1.00**	**1.00**
	MPE	6.54	13.01	69.22	**1.73**	8.36	1.89	31.55	8.33

Table 4.3. Results achieved on synthetic datasets with ID $\leq$ 10 for 9 state-of-the-art ID estimators and the one we propose (NNK). The MPE achieved by each algorithm is reported in the bottom row. For each dataset, the best approximations are highlighted in boldface.

an ID ≤ 10 are summarized in Table 4.3, those with ID > 10 are in Table 4.4, and those for the real datasets are in Table 4.5. The NNK algorithm correctly estimates 10 out of the 16 synthetic datasets, and 2 out of 2 real datasets. While very good estimates on manifolds with low ID are obtained, the NNK algorithm tends to underestimate the dimension of datasets of high ID, heavily penalizing the overall MPE score, as we hypothesized earlier when discussing the size of the NNK neighborhoods as a function of ID. Given that an exponentially increasing number of points is necessary to accurately represent the geometry of the manifold, the chosen $K = 100$ is not enough to learn high-dimensional manifolds. Regarding the real datasets, NNK is the best-performing method.

Our experiments show promising results for our NNK ID estimator, and complete execution of the estimator benchmark, i.e., with the full set of datasets and evaluation metrics, which would be interesting to better understand its performance, is left for future work.

4.2 Manifold Understanding with NNK graph properties

We have used low-rank approximations obtained from NNK neighborhoods to estimate the local ID of a set of points in a manifold. NNK graphs are built such that they give information on the geometry of the local neighborhood. We can compare different

Dataset	d	MLE	kNNG	BPCA	Hein	CD	DANCo	MLSVD	NNK
$\mathbf{M}_9$	*20*	14.64	10.59	13.55	15.50	13.75	19.71	**20.00**	10.00
$\mathbf{M}_{10b}$	*17*	12.87	15.38	9.46	13.85	12.39	16.62	**17.00**	**17.00**
$\mathbf{M}_{10c}$	*24*	16.97	21.42	13.3	17.95	15.58	24.28	**24.00**	23.00
$\mathbf{M}_{10d}$	*70*	36.96	40.31	71.00	38.69	31.4	70.52	**70.00**	29.00
$\mathbf{M}_{12}$	*20*	15.82	24.89	13.7	15.00	11.26	19.90	**20.00**	11.00
	MPE	29.69	26.84	30.22	27.19	38.46	1.22	**0.0**	31.54

Table 4.4. Results achieved on synthetic datasets with ID > 10 for 9 state-of-the-art ID estimators and the one we propose (NNK). The MPE achieved by each algorithm is reported in the bottom row. For each dataset, the best approximations are highlighted in boldface.

Dataset	d	MLE	kNNG	BPCA	Hein	CD	DANCo	MLSVD	NNK
$\mathbf{M}_{\text{ISOMAP}}$	*3.00*	4.05	4.32	4.00	**3.00**	3.37	4.00	1.00	**3.00**
$\mathbf{M}_{\text{MNIST1}}$	*8.00-11.00*	10.29	**9.58**	11.00	8.00	6.96	9.98	1.00	10.00
	MPE	12.67	22.42	24.56	7.89	19.53	19.19	78.07	**2.63**

Table 4.5. Results achieved on real datasets. The MPE achieved by each algorithm is reported in the bottom row. For each dataset, the best approximations are highlighted in boldface (when the ID takes values in a range, we have highlighted the estimates closest to the average of that range).

properties of NNK graphs to gain a better understanding of manifolds.

The sizes of NNK graphs in different positions of a manifold can give insight into the point density of a manifold, such that regions with a high density of points will have smaller graphs than sparser regions. Moreover, by comparing the relative position of NNK graphs we can assess the shape of a manifold in terms of its linearity or nonlinearity. We now take an in-depth look into how these attributes of NNK graphs can be measured and the ways they can be used to further understand the geometry of a manifold.

4.2.1 NNK graph diameter

Recall that an NNK graph is can be viewed as a convex polytope, that results from the hyperplanes associated with a node's NNK neighbors. The diameter of an NNK polytope is defined as the maximum distance between points in an NNK neighborhood:

$$d = \max_{i,j \in \mathcal{S}} \|\boldsymbol{x_i} - \boldsymbol{x_j}\|, \tag{4.2}$$

where $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are the features of points belonging to a node's NNK neighborhood $\mathcal{S}$. Given that NNK will select the first point of each direction in space, we can assess the point density of a region in a manifold by the diameter of the polytopes corresponding to its NNK graphs.

Experiment

To illustrate this, in Fig. 4.6 we show the distribution of the diameter of the NNK polytopes on the $\boldsymbol{M}_5$ manifold dataset (see Table 4.1) using the 2500 points and with a sample of 500. Notice that on the denser dataset the polytopes have a smaller diameter, while on the sparser dataset the points are further apart, making the polytopes larger.

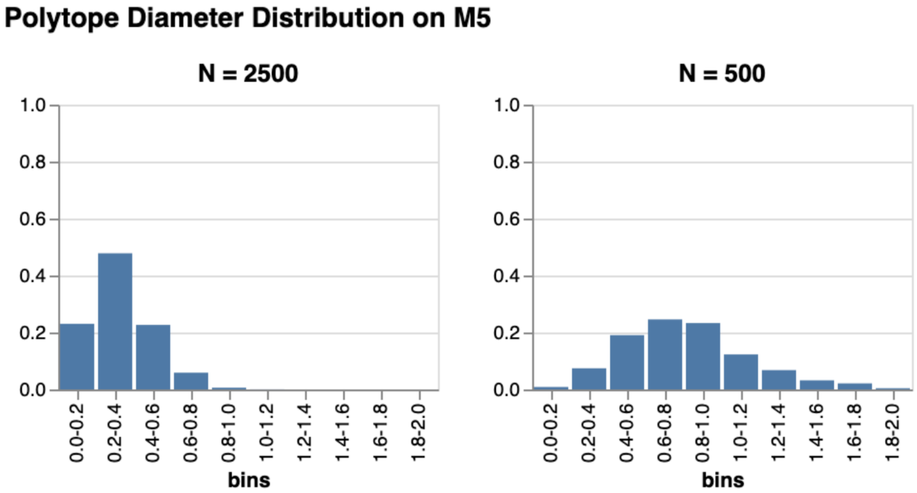

Figure 4.6. Distribution of the diameter of NNK polytope on the $\mathbf{M}_5$ manifold dataset with 2500 points and on a sample of 500. Notice that the polytopes are smaller on the denser dataset, while they become bigger when the dataset is sparse.

4.2.2 Principal angles between NNK graphs

Principal angles [29] refer to the generalization of the concept of angles between lines in the plane to any arbitrary dimension. Given two orthonormal matrices $\mathbf{A}$, $\mathbf{B}$, principal angles were computed using the Singular Value Decomposition (SVD) of the matrix $\mathbf{A}\mathbf{B}^\top$, where the eigenvalues correspond to the cosines of the principal angles. This makes it impossible to find small angles accurately on software code due to rounding errors. This issue is solved by using a sine-based approach [8] only for angles below $\pi/4$ [34]. This implementation is the one we used in our experiments.

By comparing the principal angles between the low-rank approximation of NNK neighborhoods we can better understand the geometry of a manifold. This way, on a flat (linear) manifold the distribution of the angles will be similar on neighborhoods in different positions of the manifold, and many will be close to zero. On a highly curved (nonlinear) manifold, the distribution of the angles between NNK subspaces will change at different regions in the manifold, and the angles will be higher. Moreover, on locally smooth manifolds, we expect to see small angles between the low-rank approximation of adjacent NNK neighborhoods (i.e., neighborhoods of points where one is in the NNK neighborhood of the other).

Experiment

We show this by comparing the principal angle distributions between pairs of NNK neighborhoods from different locations in a manifold on a linear and a nonlinear dataset. We compare the distribution of the angles for pairs of adjacent NNK neighborhoods and pairs of neighborhoods chosen at random. Figures 4.7 and 4.8 show the results for the linear and nonlinear manifold. respectively. For the linear manifold, the distribution of both adjacent and random pairs of neighborhoods are almost the same since the geometry of any neighborhood in a linear manifold will be similar. In contrast, on the nonlinear manifold, we see a difference in the distribution of the angles.

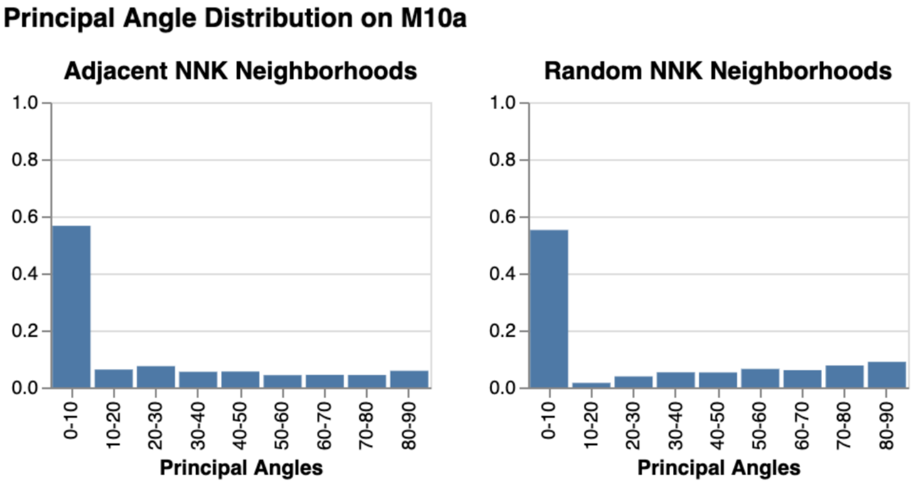

Figure 4.7. Distribution of the principal angles between NNK low-rank approximation subspaces on the linear manifold $\mathbf{M}_{10a}$ (see Table 4.1). To the left, we show the angles between pairs of adjacent NNK neighborhoods (i.e., one of the center nodes is a neighbor of the other). To the right, we show the angles for random pairs of NNK neighborhoods. On a linear manifold, any pair of neighborhoods will have angles close to 0.

4.3 Manifold Understanding with Multi-scale NNK

In the literature, multi-scale analysis is used in the context of finding an appropriate neighborhood over which to estimate the local ID. The goal is then to find a sparsity parameter (K for KNN, ε for ε-neighborhoods) such that the set of points chosen is large enough that there are at least $K \geq ID$ neighbors, small enough that the manifold is linear and large enough so that the effects of noise are negligible.

While this is a valid use of multi-scale analysis of a manifold, we hypothesize that more insight into the manifold could be gained (i.e., besides the ID) by comparing the geometrical properties of NNK graphs at different scales. Measures derived from NNK graphs can help in understanding the manifold's shape.

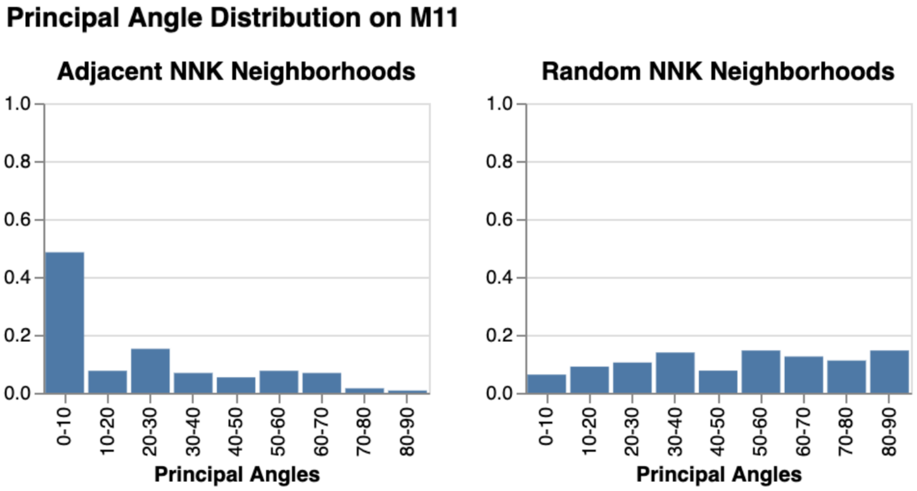

Figure 4.8. Distribution of the principal angles between NNK low-rank approximation subspaces on the nonlinear manifold $\mathbf{M}_{11}$ (See Table 4.1). To the left, we show the angles between pairs of adjacent NNK neighborhoods (i.e., one of the center nodes is a neighbor of the other). To the right, we show the angles for random pairs of NNK neighborhoods. On a nonlinear manifold, close neighborhoods have angles close to zero, while random neighborhoods are more dissimilar.

4.3.1 Multi-scale NNK algorithm

To study the manifold at larger scales, we should increase the size of the NNK neighborhood. Intuitively, we would do so by adjusting the hyperparameters of the NNK algorithm to observe points that are further away. By definition, NNK will choose a single neighbor in each relevant direction, therefore increasing the sparsity parameter K should not affect the NNK neighborhood. Alternatively, we might increase the bandwidth of the Gaussian kernel σ, but as we showed in Section 4.1.3, increasing σ will make the distance matrix ill-conditioned, since close points will be at similarity very close to 1. Instead, we propose to merge the closest points such that as we merge, distances between the points increase. We, therefore, change the scale of the analysis by making the manifold sparser, increasing the distances between points in the process (see Fig. 4.9).

While we would expect a linear manifold to look the same at different scales, if we were to perform merging on a highly curved manifold we will at some point be selecting points that initially lay on different local neighborhoods, changing the shape of the manifold in the process. This change would be reflected in the NNK graphs. By studying their geometric properties introduced in the previous section, we can attempt to better understand the shape of the manifold.

We can achieve a sparser representation by iteratively merging the two closest points according to some similarity metric. This can be based on the KNN's shortest pairwise distance, or on the largest NNK pairwise weights. We address the differences in the choice of similarity in the next section. Two points are merged by averaging their positions. Then, assuming we have a criterion for merging, in the process of merging we are

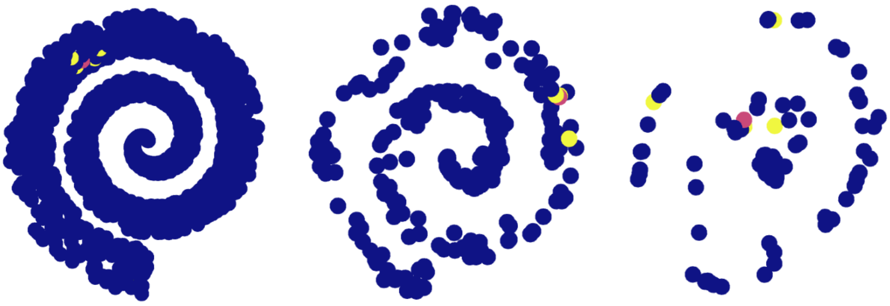

Figure 4.9. Three examples of the swiss-roll dataset, that differ in the number of points sampled (from left to right, we have 2500, 1000, and 100 points). NNK builds a graph based on the distance of a point to its neighbors, such that the size of the neighborhood is a function of the point density. This way, the scale at which we look at the manifold increases with the size of the NNK neighborhood.

eliminating points that are too close so that we can increase the window of observation. After each merging iteration, we recalculate the NNK graph. Since the decay parameter σ has been defined based on the distances in the dataset, we will in turn increase the size of the NNK graph as we merge points (see Fig. 4.10). This way, we can construct larger NNK graphs and thus be able to analyze the manifold at different scales. This merged dataset can be achieved as described in Algorithm 2.

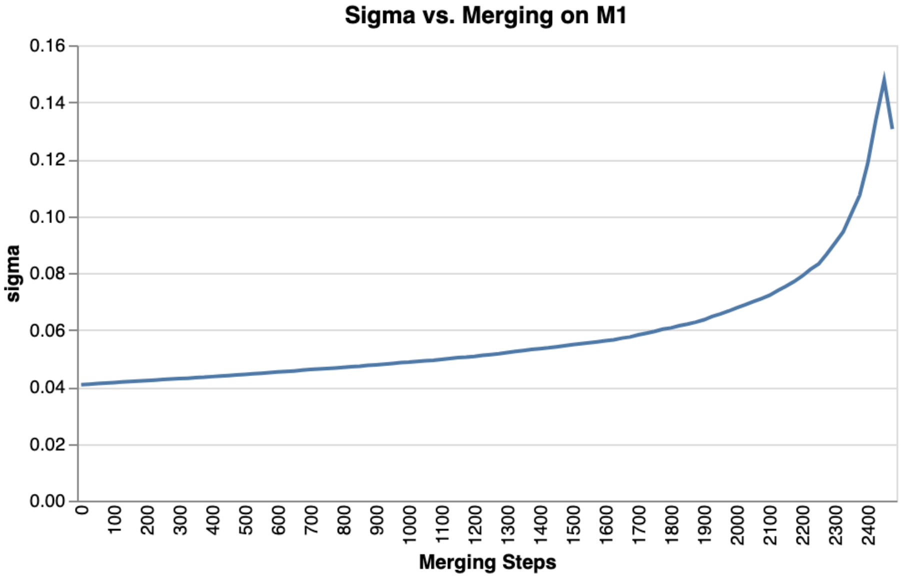

Figure 4.10. Value of σ, which is defined as $\frac{1}{3}rd$ of the 15th neighbor distance, after each merging step. See that after merging points the dataset becomes sparser and distances increase, making σ larger in the process.

The similarity metric used to select the closest points has a significant impact on the

Algorithm 2 Two-closest Merging

```
Input:
    $X$: features
    $I$: merging steps
for iter in $I$ do
    for each node $i = 1,2,...,N$ do
        $S_i = \{\text{neighborhood of node } i\}$
        $K_{i,S} = \{\text{similarity to each neighbor}\}$
    end for
    $i, j = \{i, j : \max_{i,j} K_{i,j}\}$
    $X = X \cup \frac{X_i + X_j}{2} \setminus X_i, X_j$
end for
Output: Dataset after $I$ merging steps $X$
```

point density of the manifold as we merge. We take an in-depth look at this in the next section.

4.3.2 Merging and the choice of similarity

For each node in a dataset, we will have constructed two weighted graphs: a KNN and an NNK graph. On the one hand, the KNN graph consists of K neighbors connected by edges weighted based on Euclidean distance to the center node. On the other hand, the NNK graph has edges the weight of which is the result of the NNK optimization on the initial KNN neighborhood. The weight of these edges in relation to the edges of other local connected neighborhoods in the dataset will depend on the choice of σ, such that a value of σ_{local} is defined for each neighborhood based on the distances in it, the weights will reflect the similarity in the context of the neighborhood. When a single value σ_{global} is defined for all the datasets, the pairwise similarity will be heavily influenced by the point density and not so much by the local geometry of the data (see Fig. 4.11).

There are clear differences between the similarities defined in σ_{local} NNK graphs, KNN graphs, and σ_{global} NNK graphs. The differences will be reflected in the datasets resulting from merging based on these similarities. Choosing the closest points based on KNN similarity, i.e., Euclidean distance, after some merging iterations we expect the manifold to be uniformly distributed, given that points in dense areas will be merged at a higher rate. We believe that merging based on σ_{local} NNK similarity will merge at the same rate regardless of density, given that the NNK similarity will only depend on the local distribution of the data. Merging with σ_{global} NNK similarity may show a behavior somewhere between the other two approaches. We assess this hypothesis in the following experiment.

Experiment

We will be using the distribution of the diameter of the polytopes defined by NNK graphs as a proxy for the point density in the dataset. We have tested our merging

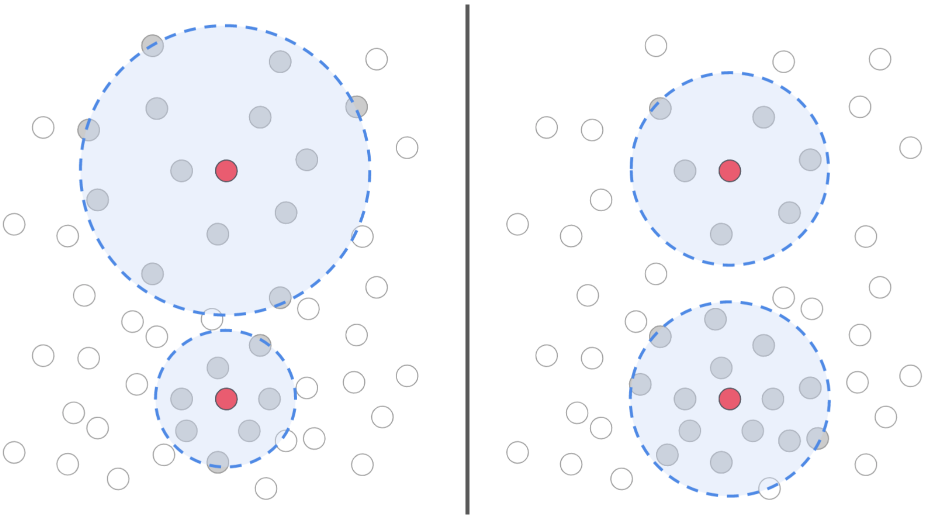

Figure 4.11. Comparison of the reach (blue circle) of the Gaussian kernel on two NNK neighborhoods using a local definition of σ (Left) and a global definition (Right). In the local case, distances in kernel space will be related to the distribution of the points in the neighborhood. In the global case, the reach of the Gaussian kernel is the same regardless of the point density, so points will be closer in kernel space in denser regions.

algorithm using KNN similarity, σ_{global} NNK similarity, and σ_{local} NNK similarity first on a *control* dataset and later on MNIST (see Table 4.2). The former allows us to control the conditions of the dataset so that we can validate our hypothesis. The latter is used to assess our algorithm on a real dataset. The control dataset is of size $N = 300$ and consists of two isolated square regions in a two-dimensional space, such that one has three times as many points as the other. For the experiments on this dataset, we use $K = 30$ and σ equal to one-third of the 15th neighbor distance (the average for all the neighborhoods in the global case). We merge one point at each iteration. In the case of MNIST, we sample 2500 from the dataset, and merge 250 points at each step for a total of 100 steps, merging the 250 closest pairs of points at each iteration without repeating any points. This approximation has a very similar behavior to merging one pair of points at each step, and it allows us to perform merging in a shorter amount of time.

Fig. 4.12 shows the number of points in the dense and sparse regions in the *control* dataset after each merging step. Using KNN merging, the dense region goes down in size much faster than the sparse region. That is until they have the same density, after which they have points merged at the same rate. Using σ_{local} NNK merging, dense and sparse regions have points merged at a similar rate until the latter has no more points left. With a global choice of σ_{global}, we see similar behavior to that of KNN. This suggests that the initial KNN weights have a significant impact on those obtained after the NNK optimization, and that data locality in the kernel similarity matrix is compromised.

Looking at the distribution of the diameter of the NNK polytopes for the same dataset, in Fig. 4.13, we observe that when using KNN similarity for merging, the size of the polytopes increases, such that the size distribution shifts to larger polytopes while also growing

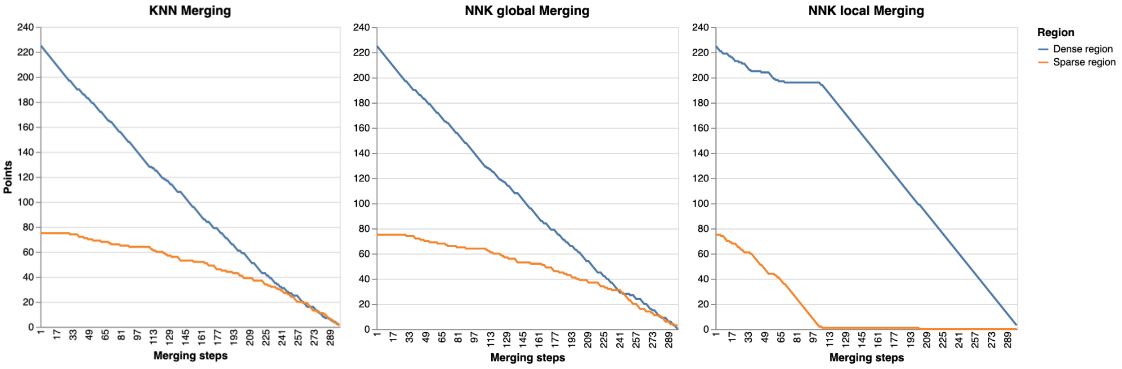

Figure 4.12. Number of points in the dense and sparse regions of the *control* dataset after each merging step using KNN similarity, σ_{global} NNK similarity, and σ_{local} NNK similarity. In the first two cases, points in dense areas are merged at a higher rate than in sparser areas. With σ_{local} merging, points in areas of different densities are merged at the same rate.

in size. In contrast, when σ_{local} NNK similarity is used for merging, the distribution of the polytope diameters is preserved as their sizes grow overall.

Finally, in Fig. 4.14 we show the distribution of the NNK polytopes for the MNIST dataset. Again, we see that as we merge with KNN similarity the polytopes increasingly grow in size, while using σ_{local} NNK similarity in the merging process the distribution of the polytope relative sizes is preserved. This suggests that the relative density of the different regions in the sparser merged dataset is the same as that of the initial one.

We have shown that by merging based on the closest points in σ_{local} NNK similarity, we can obtain a smaller sample of a dataset while maintaining the distribution of the points. This way, we can sample points preserving the geometrical properties of the original dataset. This can be useful for tasks that require the creation of a representative dataset that has a much smaller number of samples. Some applications such as the stochastic gradient descent step in the training of deep neural networks can achieve good results by sampling randomly. This is possible due to the lack of limitations on the data, which leads to all the random choices aggregating to a meaningful result. On many other tasks, sampling randomly may negatively affect the results, and it becomes necessary to build a subsampled dataset carefully. Creating a smaller dataset by merging points based on NNK similarities allows us to preserve some of its geometrical properties.

Moreover, conditions can be set to the properties of NNK graphs to stop the merging if some specific conditions are met. For example, a lower bound can be set on the polytope diameter to ensure a minimum point density in the sampled dataset.

4.4 Manifold Understanding Experiments

In the previous section, we have presented a set of tools based on the properties of NNK graphs that can help us understand manifold data. In this section, we evaluate the

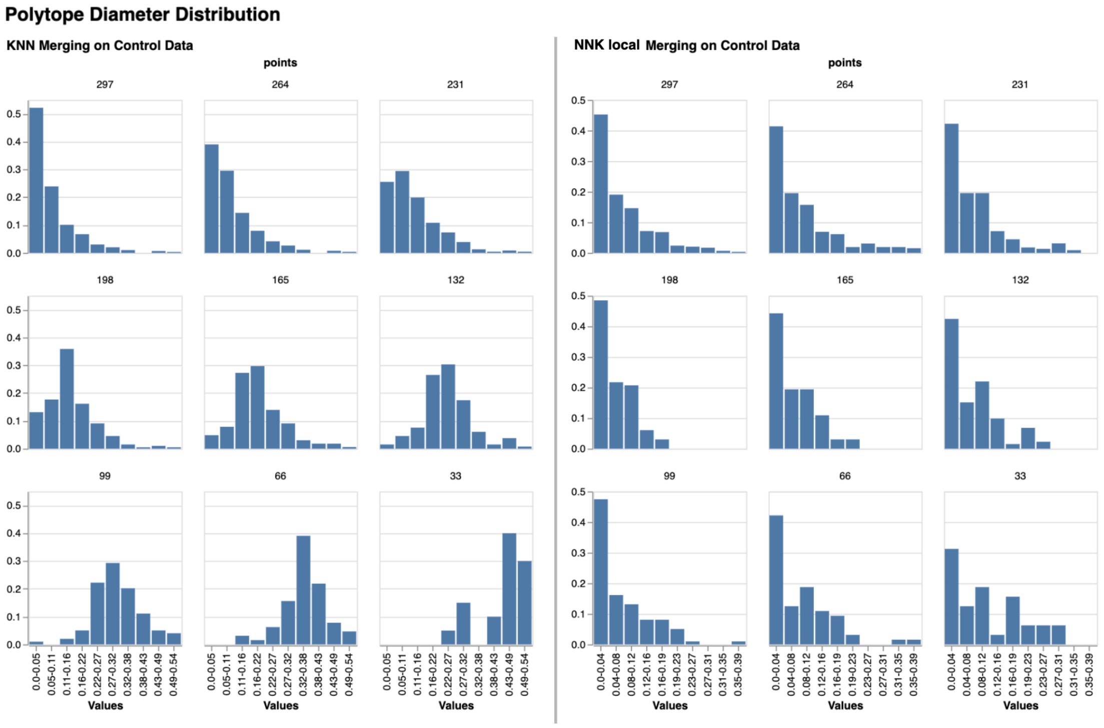

Figure 4.13. Distribution of the diameter of NNK graph polytopes constructed on the *control* dataset while merging either with KNN and σ_{local} NNK similarity. When merging the closest points in KNN similarity at each step, we observe the polytope diameter distribution shift to larger diameters. When using σ_{local} NNK similarity in the selection of the points to merge, the distribution of the polytope sizes remains the same after merging.

properties of NNK graphs we presented on top of a variety of datasets at multiple scales, using our proposed merging algorithm.

4.4.1 Experiments on synthetic datasets

We will first study the synthetic datasets (see Table 4.1), for which the geometric properties are known. In our first experiment, we assess the linearity of a manifold by looking at the number of NNK neighborhoods at different scales and applying our merging algorithm. Next, we compare the dimension of the low-rank approximations of the KNN and NNK neighborhoods to study the differences at multiple scales.

We show the metrics (i.e., number of neighbors and low-rank approximation size) in the NNK graphs constructed on the datasets after each σ_{global} merging iterations. The initial size of the datasets if of $N = 2500$ points, and at each merging step features the closest 250 pairs of points (without repetition) are averaged until no points are left. The value shown for the number of neighbors consists of the average and standard deviation for the neighborhood sizes of a sample of 300 points (or the whole dataset when the size is smaller). We have run this process on 4 linear manifolds and 4 nonlinear manifolds. The

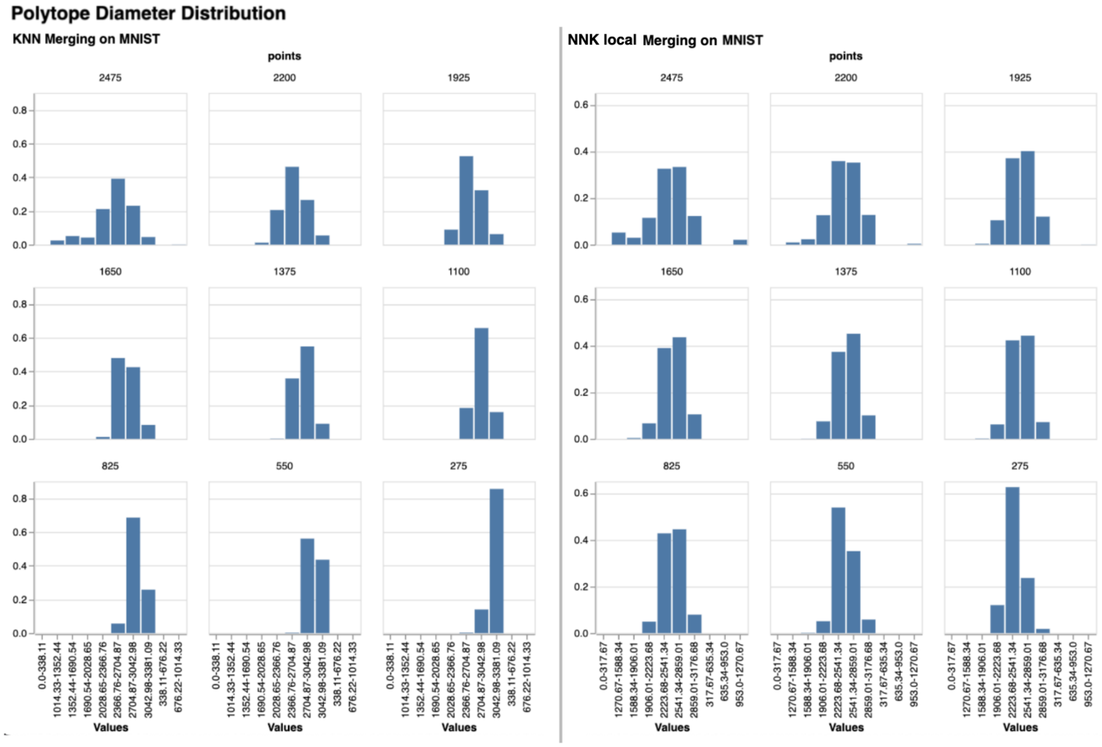

Figure 4.14. Distribution of the diameter of NNK graph polytopes constructed on the MNIST dataset while merging either with KNN or σ_{local} NNK similarity. When merging the closest points in KNN similarity at each step, we observe the polytope diameter distribution shift to larger diameters. When using σ_{local} NNK similarity in the selection of the points to merge, the distribution of the polytope sizes remains the same after merging.

linear manifolds are $\mathbf{M}_1$ (10-dimensional sphere linearly embedded), $\mathbf{M}_2$ (affine space), $\mathbf{M}_9$ (affine space), and $\mathbf{M}_{10a}$ (10-dimensional hypercube). The nonlinear manifolds are $\mathbf{M}_3$ (concentrated 4-dimensional figure), $\mathbf{M}_4$ (nonlinear manifold), $\mathbf{M}_7$ (Swiss-Roll), and $\mathbf{M}_{11}$ (Möebius band 10-times twisted).

NNK Neighborhood size at different scales

Fig. 4.15 shows the number of NNK neighbors as a function of the points left in the dataset for the linear manifolds. The number of neighbors remains consistent until the number of points has been reduced by a factor of 4 or more (600 points left). On a linear manifold, we expect the same geometry regardless of scale, and this is what we observe for the size of the NNK neighborhoods. After most of the points have been merged, the size of the neighborhoods decreases. This is both because there are fewer points, and because the manifolds show some curvature at a larger scale, such as $\mathbf{M}_1$.

We observe (see Fig. 4.16) a clear difference in the number of NNK neighbors at different scales in the case of the nonlinear manifold datasets. The number of neighbors

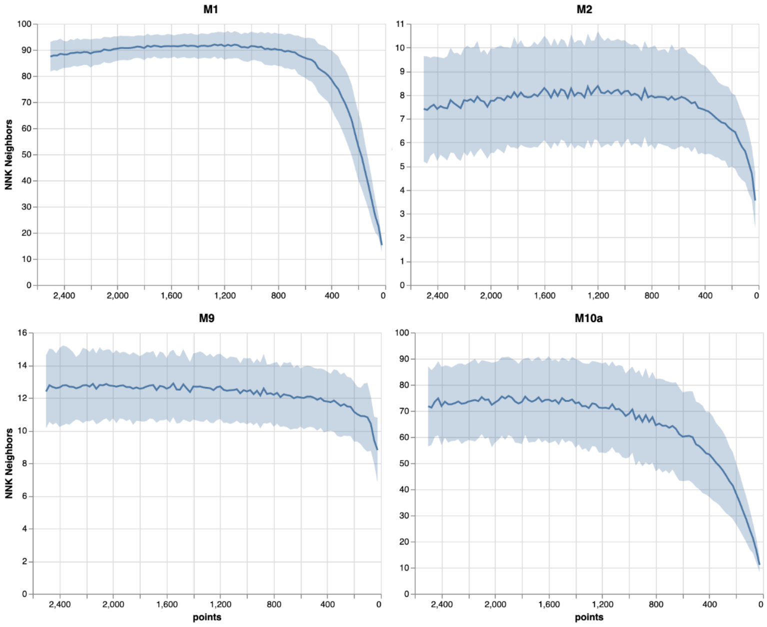

Figure 4.15. Average NNK neighbors vs. points left in the 4 linear manifold datasets after merging based on σ_{global} similarity. The neighborhood size does not change with scale.

does not remain the same but instead changes with scale. How the size of an NNK neighborhood changes depends on the geometry of the manifold. Both $\mathbf{M}_7$ and $\mathbf{M}_{11}$ correspond to 2-dimensional flats arranged in a 3-dimensional space. As scale increases, the two-dimensional surface resembles a three-dimensional volume, as depicted in Fig. 4.9. When the geometry of the manifold is lost, points outside of the initial local neighborhood are selected for the NNK graph.

KNN and NNK low-rank approximations at different scales

In Algorithm 1 we used the number of principal components obtained from the NNK neighborhoods, such that the components chosen are those with an eigenvalue $\lambda_j \geq \frac{1}{10} \cdot \lambda_{max}$. We now calculate the low-rank approximation both for the KNN and the NNK neighborhood.

On the linear manifolds, as shown in Fig. 4.17, the dimension of the low-rank approximation obtained from the KNN neighborhood (blue) is much closer to the ID of the manifold than that of the NNK neighborhood. Given that a KNN neighborhood has multiple points in each direction, each direction is learned more robustly. In contrast, NNK has a single example in each of those directions, the reason for which we observe a higher standard deviation for the dimension of the low-rank approximation. For the case of $\mathbf{M}_1$ (10-dimensional sphere shell), we can observe that as the manifold becomes sparse enough, both KNN and NNK neighborhoods select points in all directions. This is reflected in the increase in the size of the low-rank approximations to the embedding dimension of the data (i.e., the dimension of the dataset itself).

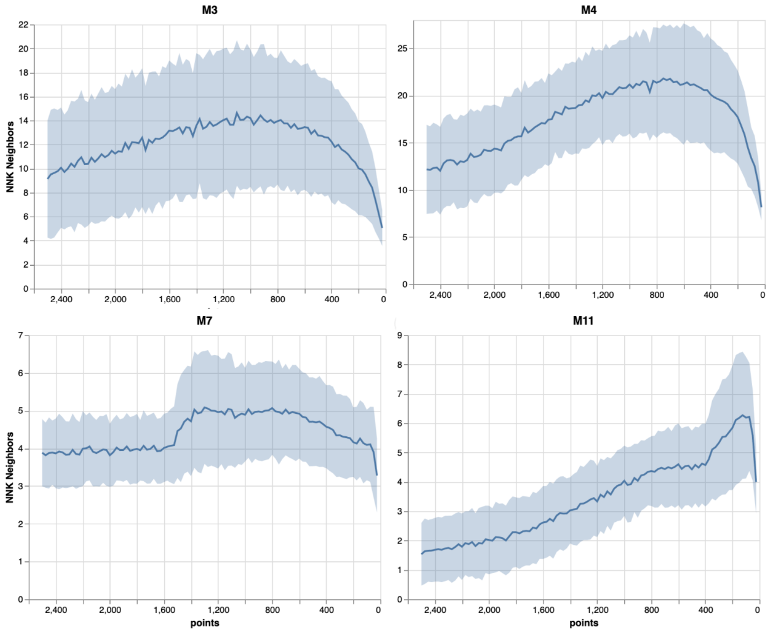

Figure 4.16. Average NNK neighbors vs. points left in the 4 nonlinear manifold datasets after merging based on σ_{global} similarity. The neighborhood size does not change with scale.

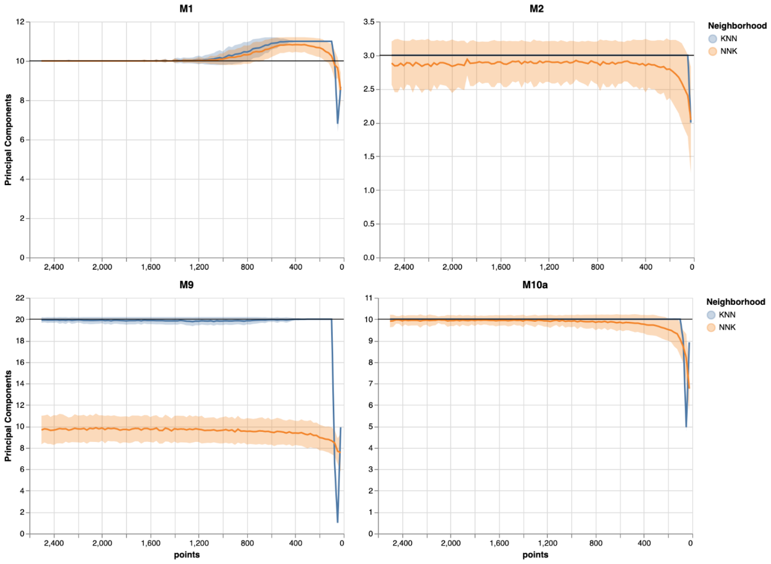

Figure 4.17. Principal components chosen for the low-rank approximation of the KNN and NNK neighborhoods as a function of the points left in the 4 linear manifold datasets after merging based on σ_{global} similarity. We also show a black line with the ID of each dataset.

Having many points in each direction will yield good results when each direction is relevant, as is the case on a linear manifold. For nonlinear manifolds this is not the case and, as depicted in Figure 4.18, low-rank approximations from KNN neighborhoods overestimate the dimension of the manifold. Furthermore, after a sufficient number of merging steps is performed such that the geometry of the manifold is lost, the low-rank approximation from KNN will be of a size equal to the embedding dimension of the data. $\boldsymbol{M}_4$ is embedded in an 8-dimensional space, and $\boldsymbol{M}_7$ and $\boldsymbol{M}_{11}$ in a 3-dimensional one. This occurs because the KNN neighborhood contains points in every direction.

The low-rank approximation derived from NNK does a better job of preserving the local geometry of the manifold. While the variance is much higher due to the sparsity of the NNK graphs relative to the KNN ones, the size of the low-rank approximation is much closer to the ID, especially in the $\boldsymbol{M}_3$ and $\boldsymbol{M}_7$ datasets.

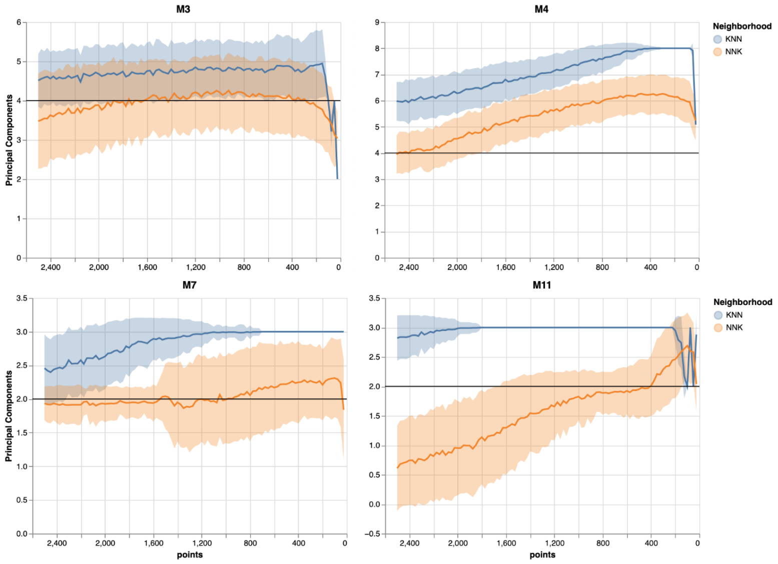

Figure 4.18. Principal components chosen for the low-rank approximation of the KNN and NNK neighborhoods as a function of the points left in the 4 nonlinear manifold datasets after merging based on σ_{global} similarity. We also show a black line with the ID of each dataset.

4.4.2 Experiments on neural network features

To understand how neural networks learn, there is work exploring the similarities between artificial vision and biological vision in terms of the architecture of the object recognition path [58]. Class separability and invariance are vital components of this process, and based on this understanding, a hypothesis was presented in an opinion paper [20] in the field of neuroscience, that can be generalized to that of machine learning. In this paper, they propose that each object is embedded in a manifold, such that in the input layer (i.e., pixel space), the manifolds for each class are highly curved and tangled, and as one moves forward in the network, the manifolds become flatter, linearly separable, and of lower dimensionality. It can be shown that the data is indeed linearly separable, given the performance of neural networks with a linear classification layer after the last hidden layer. We attempt to increase our understanding of the change in the dimensionality of data throughout a neural network [45], [2] [50], and of the curvature of the features at different depths, through the perspective of NNK graphs.

We will be analyzing the feature vectors of a VGG-19 Deep Convolutional Neural Network (CNN) trained on the CIFAR-10 [35] dataset. This network consists of 2×64, 2×128, 4×256, and 8×512 layer depth channels with ReLU activations, and max-pooling layers after the 2nd, 4th, 8th, 12th, and 16th convolutional layers. After training the network for 200 epochs with the Adam optimizer, a learning rate of 0.1, and a batch size of 128, we select a class-balanced subset of $N = 1000$ randomly sampled points from the training dataset and analyze their feature vectors after each of the pooling layers, as well as on the input and output of the network. We do so by merging points using σ_{global} NNK similarity and constructing NNK graphs on the data after each merging iteration. We use the approximate merging algorithm (i.e., merging $N/100$ pairs of points at each step) for resource efficiency. Additionally, we also perform for 10 datasets each containing 1000 images of one of the 10 classes in CIFAR. We use a $K = 100$ and $\sigma = \frac{1}{3} \cdot$ 15th neighbor distance for the NNK graphs.

Class-balanced dataset

We first look at the estimate for the ID of the features obtained at different depths in the network. Figure 4.19 shows, for each layer, a boxplot of the sizes of the low-rank approximations obtained from NNK neighborhoods constructed on the image features, such that the feature vector of each image corresponds to a point in the dataset, and the neighborhood of an images consists of other images. The ID increases in the first layers of the network, which relates to the early layers performing low-level pre-processing and feature extraction. These representations are task-independent and arise from features irrelevant to the task. Later in the network, there is a dimensionality compression, a drop in ID, which is caused by the selection of only task-relevant features. While this is not a novel result in itself, the fact that it can be replicated using NNK speaks for its reliability as an ID estimator.

We next attempt to assess the linearity or nonlinearity of these feature vectors. We focus on the features of the last three pooling layers since those are the most relevant to the task being solved by the network. Figure 4.20 shows both the average number of NNK neighbors and the size of the low-rank approximation for the KNN and NNK neighborhoods for the last three pooling layers. The size of the NNK neighborhoods changes as points are merged, and the trend is different at each of the layers. This suggests that the feature vector spaces lie on nonlinear manifolds and that these nonlinearities are different for each layer. This is supported by the change in the size of the KNN neighborhood low-rank approximations as we merge, which is explained by the KNN neighborhood finding points in new directions as the dataset becomes sparser. NNK neighborhoods are more robust to sparsity, as shown by the little change in the size of their low-rank approximations after the 4th and 5th pooling layers in the merging process.

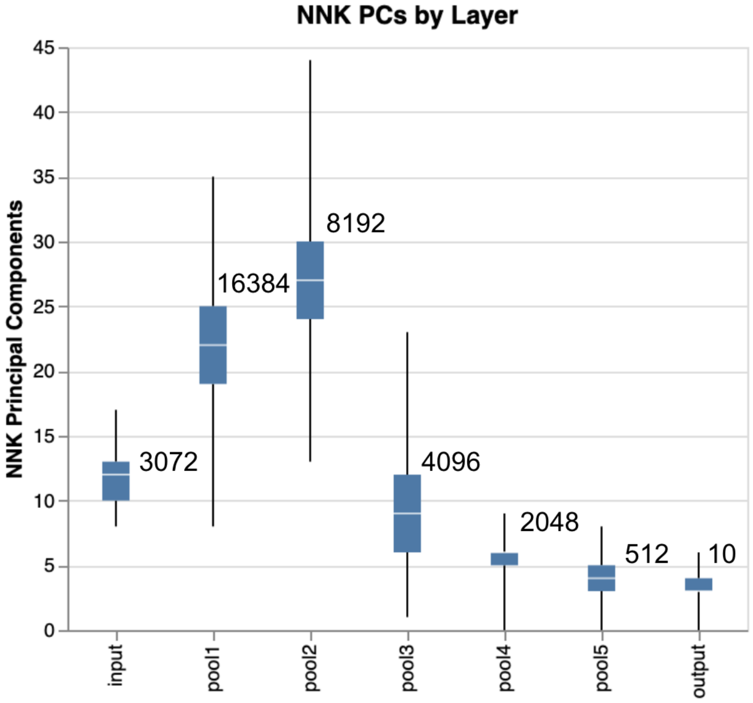

Figure 4.19. Boxplot for the dimension of the low-rank approximation of NNK neighborhoods constructed on the features of a class-balanced dataset at different depths of a VGG-19 network trained on CIFAR-10. Next to the boxplot, we show the embedding dimension of the features at each depth.

Single class datasets

We have shown that for a class-balanced dataset we observe a behavior akin to that of a nonlinear manifold. We now look at the geometry of each class independently. We do this with datasets consisting of 1000 samples for each of the classes. Our analysis is performed for the features of the samples in the last hidden layer.

Figure 4.21 shows the low-rank approximation dimension as a function of the merging steps for 6 of the 10 classes in CIFAR10, both for KNN and NNK neighborhoods. In contrast to the previous example, in this case, we observe linear behavior since the dimension of the low-rank approximations both for KNN and NNK are almost the same regardless of scale. This suggests that the features for each class lie on an almost flat manifold.

We look further into this result by comparing the angles between the low-rank approximations of adjacent and random NNK neighborhoods belonging to the same class. We show the results for 4 of the 10 classes in Fig. 4.22. The distribution of the angles between adjacent and random NNK neighborhoods is very similar, in that the majority of the principal angles are 0 to 20 degrees. This further supports the finding that the features of each class in the last hidden layer lie on a linear or almost linear manifold.

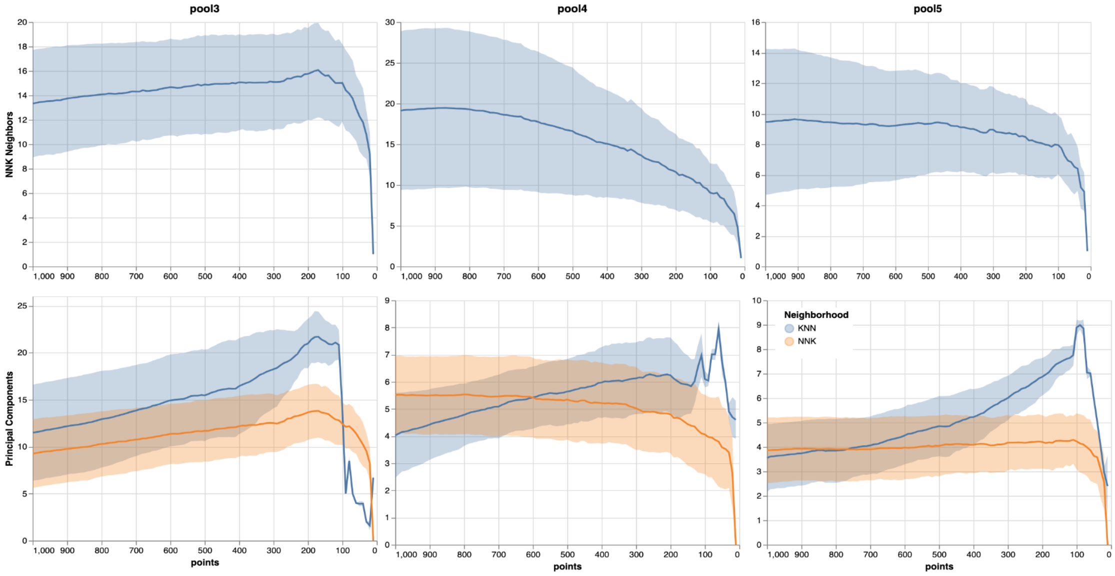

Figure 4.20. Average and standard deviation of NNK neighbors (top row) and the low-rank approximation of NNK graphs (bottom row) as a function of the remaining points in the dataset after each merging step. We show the results for the features obtained on a class-balanced dataset after the 3rd, 4th, and 5th pooling layers in a VGG-19 network trained on CIFAR-10.

To summarize, in this experiment, we have replicated the results obtained in other work in the literature regarding the intrinsic dimension of deep neural networks [45], [2] [50], approaching the estimation of ID using NNK graphs. Moreover, we have observed that the feature vector representation in the last hidden layer has the properties of a nonlinear manifold, and while previous work [2] rejects the hypothesis that neural networks flatten the representation of data in the last layers, we have observed that the features for each of the classes do lie in a linear or almost linear manifold. Further work on this topic could explore the relative position of the features between different classes. Moreover, the properties of the NNK graphs for each class could relate to their classification error.

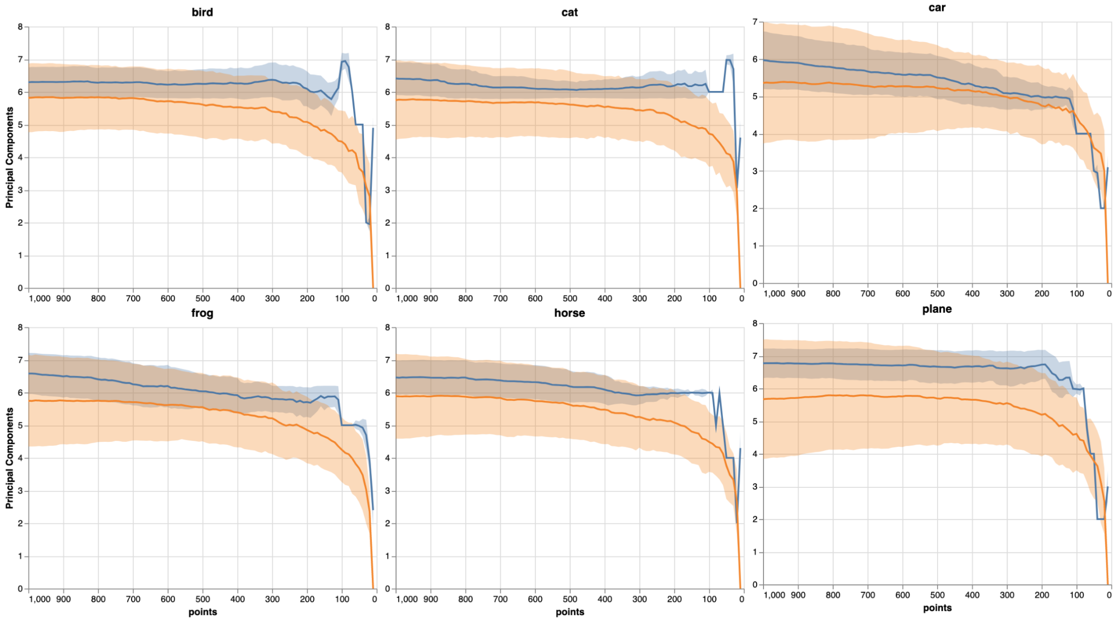

Figure 4.21. Average and standard deviation of the low-rank approximation of NNK graphs as a function of the remaining points in the dataset after each merging step. We show the results obtained for the *bird*, *cat*, *car*, *frog*, *horse*, and *plane* classes in CIFAR10. While we don't show the remaining classes due to space limitations, we observe the same behavior shown for these.

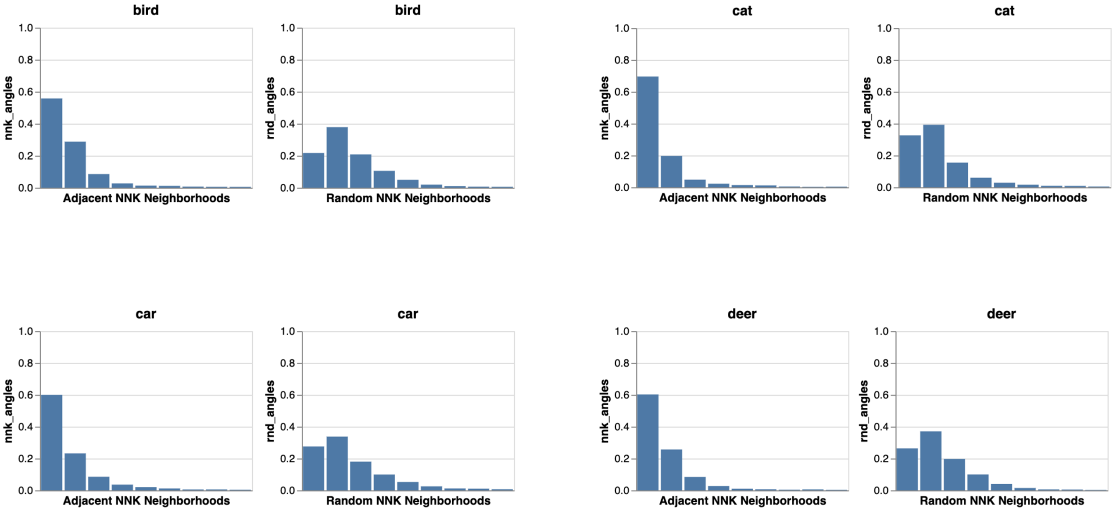

Figure 4.22. Principal angles between the low-rank approximations of adjacent NNK neighborhoods and random NNK neighborhoods for each class. The angles span between 0 and 90 degrees and are discretized in 10 uniformly distributed bins. We show the results obtained for the *bird*, *cat*, *car*, and *deer* classes in CIFAR10. While we don't show the remaining classes due to space limitations, we observe the same behavior shown for these.

Chapter 5

Conclusion and Future Work

The main goal of this project was to exploit the geometrical properties of NNK graphs constructed by optimizing an initial KNN neighborhood, to gain insight into the shape of data manifolds. We have developed a toolkit based on the properties of NNK graphs that allows us to gain insight into the geometry of data manifolds in terms of their intrinsic dimension, curvature, and point density. The capabilities and limitations of NNK graphs in this context have also been discussed. The proposed metrics are the number of NNK neighbors, the dimension of the low-rank approximation of the KNN and NNK graphs, the diameter of NNK graphs, and the principal angles between the low-rank approximations of NNK graphs. Moreover, we compare these metrics at multiple scales, which we can do by using our proposed point merging algorithm.

We have evaluated our ID estimator and compared it to other state-of-the-art estimators on an ID estimator benchmark. Moreover, we have shown the effectiveness of these tools in estimating dimension and curvature on a variety of datasets with known properties. Additionally, we have also gained insight into unknown manifolds by applying the tools described.

We have also shown that by merging based on the closest points in NNK similarity, we can obtain a smaller sample of a dataset while maintaining the distribution of the points. This way, we can sample points preserving the geometrical properties of the original dataset.

Future work should further assess the quality and reliability of our NNK ID estimator by performing the full ID estimator benchmark. This would allow us to better understand how NNK neighborhoods are selected in relation to the data. Furthermore, we have used the tools proposed to determine the dimension, point density, and linearity or nonlinearity of a manifold as a whole. By looking at these properties for different NNK neighborhoods that are at a measurable distance, such as the Euclidean distance between the center nodes, or distance in hops away from a node in NNK graphs. This would allow us to better understand the shape of the manifold at different regions which, as we saw in the last hidden layer of the VGG-19 network, can change based on location.

We discussed the need for a multi-scale approach to graph construction due to the possibility of noise in the data, but we did not address that scenario in our experiments. Further work should look into how merging can be leveraged to reduce noise in the data, in a behavior similar to a low-pass filter. The number of merging steps required to remove the noise from a dataset while preserving its geometry is not trivial and should be looked into. In the topic of merging, future work should consider merging based on properties of NNK graphs other than the neighbor similarity. This can result in dataset subsamples with other possibly interesting properties.

Finally, our approach to dataset subsampling while preserving the initial dataset properties could provide useful in tasks that rely on random sampling. The merging process is deterministic, and the result of merging after a fixed number of iterations will not change, but for each point in the merged dataset, we have knowledge of the points in the original dataset that were used to arrive at it. This information could be leveraged to train a neural network on batches generated by merging a dataset to a size equal to the batch size and sampling a point from each of the clusters in the merged dataset. With this approach, every batch would have samples in a way that the geometry of the training data is preserved.

Bibliography

[1] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[2] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Jonathan Bac and Andrei Zinovyev. Local intrinsic dimensionality estimators based on concentration of measure. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[4] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[5] Robert Bennett. The intrinsic dimensionality of signal collections. *IEEE Transactions on Information Theory*, 15(5):517–525, 1969.

[6] Christopher Bishop. Bayesian pca. *Advances in neural information processing systems*, 11, 1998.

[7] Christopher M Bishop et al. *Neural networks for pattern recognition.* Oxford university press, 1995.

[8] ke Björck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973.

[9] David Bonet, Antonio Ortega, Javier Ruiz-Hidalgo, and Sarath Shekkizhar. Channel redundancy and overlap in convolutional neural networks with channel-wise nnk graphs. *arXiv preprint arXiv:2110.11400*, 2021.

[10] David Bonet, Antonio Ortega, Javier Ruiz-Hidalgo, and Sarath Shekkizhar. Channel-wise early stopping without a validation set via nnk polytope interpolation. *arXiv preprint arXiv:2107.12972*, 2021.

[11] MR Brito, Adolfo J Quiroz, and Joseph E Yukich. Intrinsic dimension identification via graph-theoretic methods. *Journal of Multivariate Analysis*, 116:263–277, 2013.

[12] Jörg Bruske and Gerald Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on pattern analysis and machine intelligence*, 20(5):572–575, 1998.

[13] Francesco Camastra. Data dimensionality estimation methods: a survey. *Pattern recognition*, 36(12):2945–2954, 2003.

[14] Paola Campadelli, Elena Casiraghi, Claudio Ceruti, and Alessandro Rozza. Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015, 2015.

[15] Lawrence Cayton. Algorithms for manifold learning. *Univ. of California at San Diego Tech. Rep*, 12(1-17):1, 2005.

[16] Claudio Ceruti, Simone Bassis, Alessandro Rozza, Gabriele Lombardi, Elena Casiraghi, and Paola Campadelli. Danco: dimensionality from angle and norm concentration. *arXiv preprint arXiv:1206.3881*, 2012.

[17] Václav Chvátal and PL Hammer. Aggregations of inequalities. *Studies in Integer Programming, Annals of Discrete Mathematics*, 1:145–162, 1977.

[18] Jose A Costa, Abhishek Girotra, and AO Hero. Estimating local intrinsic dimension with k-nearest neighbor graphs. In *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, pages 417–422. IEEE, 2005.

[19] Jose A Costa and Alfred O Hero. Determining intrinsic dimension and entropy of high-dimensional shape spaces. In *Statistics and analysis of shapes*, pages 231–252. Springer, 2006.

[20] James J DiCarlo and David D Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007.

[21] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

[22] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):1–8, 2017.

[23] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[24] Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 100(2):176–183, 1971.

[25] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. In *The theory of chaotic attractors*, pages 170–189. Springer, 2004.

[26] Vincent Gripon, Antonio Ortega, and Benjamin Girault. An inside look at deep neural networks using graph signal processing. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[27] Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in rd. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296, 2005.

[28] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[29] Camille Jordan. Essai sur la géométrie à n dimensions. *Bulletin de la Société mathématique de France*, 3:103–174, 1875.

[30] Ashish Kapoor, Hyungil Ahn, Yuan Qi, and Rosalind Picard. Hyperparameter and kernel learning for graph based semi-supervised classification. *Advances in neural information processing systems*, 18, 2005.

[31] Masayuki Karasuyama and Hiroshi Mamitsuka. Adaptive edge weighting for graph-based learning algorithms. *Machine Learning*, 106(2):307–335, 2017.

[32] Daniel N Kaslovsky and François G Meyer. Optimal tangent plane recovery from

noisy manifold samples. *ArXiv eprints*, 2011.

[33] Balázs Kégl. Intrinsic dimension estimation using packing numbers. *Advances in neural information processing systems*, 15, 2002.

[34] Andrew V Knyazev and Merico E Argentati. Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002.

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[36] Norbert Krüger and Michael Felsberg. A continuous formulation of intrinsic dimension. In *BMVC*, pages 1–10. Citeseer, 2003.

[37] Carlos Lassance, Myriam Bontonou, Ghouthi Boukli Hacene, Vincent Gripon, Jian Tang, and Antonio Ortega. Deep geometric knowledge distillation with graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8484–8488. IEEE, 2020.

[38] Carlos Lassance, Vincent Gripon, and Antonio Ortega. Laplacian networks: Bounding indicator function smoothness for neural networks robustness. *APSIPA Transactions on Signal and Information Processing*, 10, 2021.

[39] Carlos Lassance, Vincent Gripon, and Antonio Ortega. Representing deep neural networks latent space geometries with graphs. *Algorithms*, 14(2):39, 2021.

[40] Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

[41] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[42] Anna V Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for estimating intrinsic dimension. *Proc. SampTA*, 4(2), 2011.

[43] Anna V Little, Mauro Maggioni, and Lorenzo Rosasco. Multiscale geometric methods for data sets i: Multiscale svd, noise and curvature. *Applied and Computational Harmonic Analysis*, 43(3):504–567, 2017.

[44] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[45] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[46] Vladimir Pestov. An axiomatic approach to intrinsic dimension of a dataset. *Neural Networks*, 21(2-3):204–213, 2008.

[47] Vladimir Pestov. Intrinsic dimensionality. *SIGSPATIAL Special*, 2(2):8–11, 2010.

[48] Karl W Pettis, Thomas A Bailey, Anil K Jain, and Richard C Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on pattern analysis and machine intelligence*, (1):25–37, 1979.

[49] Phillip Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. *arXiv preprint*

arXiv:2104.08894, 2021.

[50] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

[51] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[52] Sarath Shekkizhar and Antonio Ortega. Deepnnk: Explaining deep models and their generalization using polytope interpolation. *arXiv preprint arXiv:2007.10505*, 2020.

[53] Sarath Shekkizhar and Antonio Ortega. Efficient graph construction for image representation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1956–1960. IEEE, 2020.

[54] Sarath Shekkizhar and Antonio Ortega. Graph construction from data by nonnegative kernel regression. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3892–3896. IEEE, 2020.

[55] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[56] Nakul Verma. Distance preserving embeddings for general n-dimensional manifolds. In *Conference on Learning Theory*, pages 32–1. JMLR Workshop and Conference Proceedings, 2012.

[57] Peter J. Verveer and Robert P. W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on pattern analysis and machine intelligence*, 17(1):81–86, 1995.

[58] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

[59] Peng Zhang, Hong Qiao, and Bo Zhang. An improved local tangent space alignment method for manifold learning. *Pattern Recognition Letters*, 32(2):181–189, 2011.

[60] Zhenyue Zhang, Jing Wang, and Hongyuan Zha. Adaptive manifold learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):253–265, 2011.